THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

A Lagrangian-Eulerian simulation framework for viscoelastic fluid flows

SIMON INGELSTEN

Department of Industrial and Materials Science

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2020

Acknowledgements

I would like to thank my main supervisor Roland Kádár and my co-supervisors Andreas Mark and Fredrik Edelvik. Thank you for all your support, including discussing and believing in my ideas, helping me make them come to life in the code as well the support in the writing the papers and creating this thesis. I would also like to thank Klas Jareteg for the educative discussions about GPU computations and programming, as well as my colleagues at the department of Computational Engineering and Design at the Fraunhofer-Chalmers Centre.

Finally, thank you Louise for always standing by my side.

This work has been supported in part by the Swedish Governmental Agency for Innovation Systems, VINNOVA, through the FFI Sustainable Production Technology program, and in part by the Production Area of Advance at Chalmers University of Technology. The research has been partly carried out in a Centre for Additive Manufacturing – Metal (CAM2) in a joint project financed by Swedish Governmental Agency of Innovation Systems (Vinnova), coordinated by Chalmers University of Technology. The support is gratefully acknowledged.

Abstract

Viscoelastic fluids appear in various industrial applications, including adhesive application, additive manufacturing, seam sealing and parts assembly with adhesive. These processes are characterized by complex geometry, moving objects and transient multiphase flow, making them inherently difficult to simulate numerically. Furthermore, substantial amount of work is typically necessary to setup simulations and the simulation times are often unfeasible for practical use.

In this thesis a new Lagrangian-Eulerian numerical method for viscoelastic flow is proposed. The viscoelastic constitutive equation is solved in the Lagrangian frame of reference, while the momentum and continuity equations are solved on an adaptive octree grid with the finite volume method. Interior objects are modeled with implicit immersed boundary conditions.

The framework handles multiphase flows with complex geometry with minimal manual effort. Furthermore, compared to other Lagrangian methods, no re-meshing due to grid deformation is necessary and a relatively small amount of Lagrangian nodes are required for accurate and stable results. No other stabilization method than both sides diffusion is found necessary.

The new method is validated by numerical benchmarks which are compared to analytic solutions as well as numerical and experimental data from the literature. The method is implemented both for CPU computation and in a hybrid CPU-GPU version. A substantial increase in simulation speed is found for the CPU-GPU implementation. Finally, an industrially suitable model for swirl adhesive application is proposed and evaluated. The results are found to be in good agreement with experimental adhesive geometries.

Contents

Nomenclature

Roman Symbols

$a_{n,i}$	BDF formula function value coefficients
$\mathbf{A}$	Gordon-Schowalter derivative velocity gradient terms
a_s	Convected derivative slip parameter
b_n	BDF formula time derivative coefficient
$\mathbf{B}$	RBF interpolation polynomial contribution matrix
$\mathbf{b}$	Square-root conformation tensor
$\mathbf{c}$	Conformation tensor
d	Number of spatial dimensions
d_n	Swirl nozzle diameter
$\varepsilon_{\text{neigh}}$	Node distribution neighbor tolerance
$\mathcal{Y}$	Conformation tensor equation scalar-valued function
$\boldsymbol{\mathcal{F}}$	Constitutive equation model function
f	General scalar function
$\mathbf{f}$	RBF interpolation right hand side vector
$\mathbf{g}$	Body force
G	Elastic modulus

$\mathcal{H}$	Conformation tensor equation tensor-valued function
$\mathbf{I}$	Unit tensor
l	Existensibility in the FENE models
$\mathbf{L}$	General second order tensor
L_{app}	Application distance
$\mathbf{M}$	RBF interpolation matrix
$\mathbb{N}$	Natural numbers
N_c	Number of close points in RBF interpolation
N_{cells}	Number of cells in Eulerian grid
N_{loc}	Number of local time steps in implicit Euler ODE solver
N_m	Number of viscoelastic stress modes
$n_{\max}$	Max number of Lagrangian node in a subcell
n_{split}	Number of subcells in each direction for node distribution
P	RBF interpolation polynomial term in
p	Pressure
$\mathbf{r}$	RBF interpolation position
$\mathbb{R}$	Real numbers
r_t	Swirl injection torus radius
$\mathbf{S}$	Strain rate tensor
t	Time
$\mathbf{u}$	Velocity vector
$\mathbf{u}_{\text{app}}$	Injection applicator velocity

$\mathbf{u}_{\text{flow}}$ Injection velocity based on flow rate

$\mathbf{u}_{\text{inj}}$ Total injection velocity

$\mathbf{v}$ RBF interpolation polynomial coefficient vector

$\dot{V}$ Volume flow rate

$\mathbf{w}$ RBF interpolation weight vector

$\mathbf{x}$ Position vector

$\mathbf{y}$ ODE solution vector

Dimensionless Numbers

De Deborah number

El Elasticity number

Re Reynolds number

Wi Weissenberg number

Greek Symbols

α Volume fraction in VOF method

α_G Dimensionless parameter in the Giesekus constitutive model

β Viscosity ratio

ε Non-dimensional parameter in the PTT model

ε_{tol} Numerical tolerance

η Polymeric viscosity

γ Scalar strain

$\boldsymbol{\gamma}$ Strain tensor

λ Relaxation time

λ_r	Retardation time
μ	Solvent viscosity
μ_a	Artificial viscosity
ϕ	General quantity
ψ	Radial basis function
ρ	Density
σ	Scalar stress
$\boldsymbol{\sigma}$	Total stress
$\boldsymbol{\tau}$	Viscoelastic stress
$\boldsymbol{\tau}_d$	Deviatoric stress
$\boldsymbol{\Theta}$	Log-conformation tensor
θ	Swirl injection torus angle
$\dot{\theta}$	Swirl injection torus angular velocity
ξ	Gordon-Schowalter convected derivative parameter
ζ_s	RBF interpolation scaling parameter

Acronyms

ALPM	Adaptive Lagrangian particle method
BDF	Backward differentiation formula
BLPM	Backward-tracking Lagrangian particle method
BSD	Both sides diffusion
BVH	Bounding volume hierarchy
CPU	Core processing unit

FENE Finitely extensible nonlinear elasticity

GPU Graphics processing unit

HWNP High Weissenberg number problem

LCR Log-conformation representation

LPM Lagrangian particle method

PDPS Positive definiteness preserving scheme

PTT Phan Thien Tanner (model)

SIMPLEC Semi-implicit method for pressure-linked equations - consistent

SIMPLE Semi-implicit method for pressure-linked equations

SPTT Simplified Phan Thien Tanner (model)

SRCR Square-root conformation representation

UCM Upper-convected Maxwell (model)

VOF Volume of fluid

Other

0 Zero matrix or vector

Chapter 1

Introduction

Viscoelastic fluids are common in various industrial flows, including polymer extrusion, additive manufacturing, seam sealing, adhesive application and parts assembly with adhesives. In many such processes, significant manual effort is spent to prepare the system for new products, as well as to optimize the process in terms of quality, material consumption and production cost. A large part of the preparation and optimization requires extensive physical testing. Consequently, valuable production time is lost and physical products may go to waste.

Numerical simulations offer a possible remedy for the long preparation times. If important properties of a process can be predicted through simulations, a large part of the required product and process development may be performed in a virtual environment. The amount of physical testing required may thus be reduced. Furthermore, since numerical simulations do not occupy the physical production equipment, less production time is wasted.

In this thesis a Lagrangian-Eulerian framework for numerical simulation of transient viscoelastic flow is proposed and evaluated. In the proposed method, the constitutive equation for viscoelastic stress is solved along the trajectories of Lagrangian fluid elements which are convected by the flow. The fluid momentum and continuity equations, and the transport equation for the fluid volume fraction in case of multiphase flow, are solved in the Eulerian frame of reference, discretized on an automatic and adaptive octree grid with the finite volume method. Boundary conditions on solid objects in the computational domain are treated using implicit immersed boundary conditions. The coupling between the momentum balance and the constitutive equation is realized by

interpolating the viscoelastic stress to the cell centers of the Eulerian grid using radial basis functions.

Due to the Lagrangian formulation of the constitutive equations, the calculation of the viscoelastic stresses is straightforward to parallelize and, hence, can be made highly efficient. This should be compared to solving three or six large, coupled matrix equations, respectively for 2D and 3D, which is required for a finite volume or finite element method discretization of the constitutive equation. Furthermore, no expensive re-meshing is required, as for purely Lagrangian methods, and a small amount of Lagrangian nodes per fluid cell is sufficient.

The proposed method is implemented and simulation results are validated with analytic solutions for steady and transient flow as well as with available numerical results from the literature. Furthermore, a GPU-accelerated implementation of the method is evaluated with respect to computational efficiency. Finally, a framework to model the flow of viscoelastic fluids emerging from different type of nozzles moving along arbitrary application paths is presented. A model for swirl adhesive application within the framework is proposed, implemented and evaluated.

1.1 Outline of thesis

In the following parts of this chapter, a short background of the research project is given. The concept of viscoelasticity is then introduced, followed by an overview of constitutive models for viscoelastic flow. A short review of numerical methods for viscoelastic flow simulations is then given. Finally, the software framework used to implement the method proposed in this thesis is presented.

In the second chapter, the numerical method is presented in detail. The main focus is aimed at the Lagrangian-Eulerian framework to calculate the viscoelastic stress. This is also the main scientific contribution of this thesis. A brief presentation of the Eulerian finite volume solver is also given. The chapter is concluded by a brief discussion around the implementation of the numerical method for the core processing unit (CPU) and the graphics processing unit (GPU).

In the third chapter different aspects of the proposed method are validated by comparing simulations to analytic solutions, numerical results from the literature and

experimental data. Finally, in the fourth chapter the work is concluded and the outlook on future work is discussed.

1.2 Background of research project

This thesis is part of a long-term project at the Fraunhofer-Chalmers Centre to develop CFD-based simulation tools for various key production processes. The research has been carried out over the years in both publicly and industrially funded projects. One primary focus has been numerical modeling of surface treatment processes as well as joining operations. This includes electrostatic spray painting, seam sealing, adhesive application and parts assembly with adhesive. The main contribution of this thesis is a numerical framework to perform computationally efficient simulation of the flow of viscoelastic adhesives on arbitrary geometries.

1.3 Introduction to viscoelasticity

Viscoelastic materials refer to those which simultaneously exhibit properties of both viscous fluids and elastic solids. An illustrative way to model viscoelastic material behavior is therefore through models of ideal viscous fluids and elastic solids, which constitute limiting cases of viscoelasticity. The following discussion is based on Barnes *et al.* (1989) and Bird *et al.* (1987b).

An ideal viscous fluid can be modeled as a viscous damper and an ideal elastic solid as a Hookean spring. In Figure 1.1 a sketch of a damper with viscosity η and a spring with elastic modulus G are shown.

Figure 1.1: Viscous damper (left) and Hookean spring (right).

The damper obeys Newton's viscosity law, stating that the stress is proportional to the rate of deformation with the viscosity η as

$$\sigma = \eta \frac{d\gamma}{dt}, \tag{1.1}$$

where γ is the strain in the damper. The spring obeys Hooke's law of elasticity. The stress in the spring is then proportional to the deformation through the elastic modulus G as

$$\sigma = G\gamma. \tag{1.2}$$

The two simplest ways to combine the damper and the spring to model a viscoelastic material are to connect them either in series or in parallel. If they are connected in series a Maxwell element is obtained. Conversely, if they are connected in parallel a Kelvin-Voigt element is obtained. The two elements are illustrated in Figure 1.2.

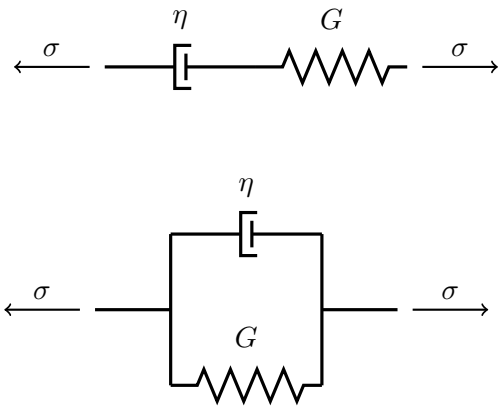

Figure 1.2: Maxwell element (top) and Kelvin-Voigt element (bottom).

For the Maxwell element, the stress is equal over the system and the strain is the sum of the strain in the damper and in the spring. Hence,

$$\sigma = \sigma_D = \sigma_S, \tag{1.3}$$

$$\gamma = \gamma_D + \gamma_S, \tag{1.4}$$

where the subscripts D and S denotes the damper and the spring, respectively. Taking the time derivative on both sides of (1.4), and inserting (1.2) and (1.1), yield

$$\frac{d\gamma}{dt} = \frac{\sigma}{\eta} + \frac{1}{G}\frac{d\sigma}{dt}. \tag{1.5}$$

By introducing the Maxwell relaxation time $\lambda = \eta/G$, the equation can be rearranged as

$$\lambda\dot{\sigma} + \sigma = \eta\dot{\gamma}, \tag{1.6}$$

where $\dot{(\bullet)}$ denotes time derivative and $\dot{\gamma}$ is called the strain rate or, in case of shear strain, shear rate.

An equation for the Kelvin-Voigt model can be derived in a similar way by realizing that the strains in the damper and the spring are equal while the total stress is the sum of the stresses in the respective elements. This leads to the equation

$$\sigma = G\gamma + \eta\dot{\gamma}. \tag{1.7}$$

The Maxwell model and the Kelvin-Voigt elements are models of linear viscoelasticity. The common property of the linear viscoelastic model equations is that the relation between the stress and the strain is described by a linear ordinary differential equation (ODE). A general such ODE can be formulated as (Barnes *et al.*, 1989)

$$\left(1 + \sum_{i=1}^{n} \alpha_i \frac{d^i}{dt^i}\right) \sigma = \left(\beta_0 + \sum_{k=1}^{m} \beta_k \frac{d^k}{dt^k}\right) \gamma, \tag{1.8}$$

where $\{\alpha_i\}_{i=1}^{n}$ and $\{\beta_k\}_{k=0}^{m}$ are constant coefficients. For example, by letting all coefficients be zero except for $\alpha_1 = \lambda$ and $\beta_1 = \eta$ the Maxwell model in the form of (1.6) is obtained. If all coefficients are zero except $\beta_0 = G$ and $\beta_1 = \eta$ the Kelvin-Voigt model is obtained. The generalization of linear viscoelasticity to constitutive equations for viscoelastic flow is discussed in Section 1.4.5.

1.4 Governing equations for viscoelastic flow

In this section, the governing equations for viscoelastic flow are discussed. This includes the momentum and continuity equations as well as constitutive models for viscoelastic flow.

1.4.1 The Lagrangian and Eulerian frames of reference

Prior to introducing the governing equations for viscoelastic flow, it is necessary to discuss different frames of reference and the relation between their corresponding time derivatives. The Lagrangian frame of reference, or the material description, describes properties of material points which move with the flow. Conversely, in the Eulerian frame of reference, or the spatial description, properties are described at fixed spatial locations.

Consider first the Lagrangian frame. Let $\mathbf{x}_0$ be the location of a material point at time $t = 0$ and its location at time t is denoted $\mathbf{x}(\mathbf{x}_0, t)$, such that $\mathbf{x}(\mathbf{x}_0, 0) = \mathbf{x}_0$. The rate of change of a property ϕ at the current location of the material point is denoted $\frac{\partial \phi}{\partial t}$. If, however, properties are described in the Eulerian frame, the rate of change must include the change of frame. The time derivative of ϕ at the location $\mathbf{x}$ in space is given by the chain rule as

$$\frac{d}{dt}(\phi(\mathbf{x}(\mathbf{x}_0, t), t) = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial t} = \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \frac{\mathrm{D}\phi}{\mathrm{D}t}, \tag{1.9}$$

where $\frac{\mathrm{D}\phi}{\mathrm{D}t}$ is called the Lagrangian derivative, or the material time derivative, of ϕ.

1.4.2 The momentum and continuity equations

Two conservation principles are important to describe the flow of a continuum, namely conservation of mass and linear momentum. Conservation of mass is expressed by the continuity equation, which for unsteady flow of a general fluid reads (Schlichting & Gersten, 2000)

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho \nabla \cdot \mathbf{u} = 0. \tag{1.10}$$

If the flow is incompressible the material time derivative of the density is zero. The continuity equation then reduces to

$$\nabla \cdot \mathbf{u} = 0, \tag{1.11}$$

which may be referred to as a divergence-free velocity field. Conservation of linear momentum is expressed by Cauchy's first law of continuum mechanics (Truesdell & Rajagopal, 1999) and can be expressed as

$$\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}, \tag{1.12}$$

where ρ is density, $\mathbf{u}$ velocity, $\boldsymbol{\sigma}$ the Cauchy stress tensor and $\mathbf{g}$ a body force. Furthermore, according to Cauchy's second law of continuum mechanics,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathsf{T}}, \tag{1.13}$$

i.e. the stress tensor is symmetric.

For a given Cartesian coordinate system $Oxyz$, $\boldsymbol{\sigma}$ can be expressed in matrix form as

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{pmatrix}. \tag{1.14}$$

It is useful to define the isotropic pressure as

$$p = -\frac{1}{3}\mathrm{Tr}\left(\boldsymbol{\sigma}\right) = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \tag{1.15}$$

and the normal stresses

$$\begin{aligned} \tau_{xx} &= \sigma_{xx} + p, \\ \tau_{yy} &= \sigma_{yy} + p, \\ \tau_{zz} &= \sigma_{zz} + p. \end{aligned} \tag{1.16}$$

The stress can then be expressed as

$$\boldsymbol{\sigma} = \begin{pmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{pmatrix} + \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{pmatrix}, \tag{1.17}$$

or in vector form as

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}_d, \tag{1.18}$$

where $\mathbf{I}$ is the identity tensor. The form of (1.18) shows the decomposition of the Cauchy stress tensor into the isotropic part, i.e. normal stresses that are equal in all directions, and the deviating part. The tensor $\boldsymbol{\tau}_d$ is therefore called the deviatoric stress (Schlichting & Gersten, 2000).

An important type of fluid is the Navier-Stokes fluid, or Newtonian fluid, which in the incompressible case has the deviatoric stress (Truesdell & Rajagopal, 1999)

$$\boldsymbol{\tau}_d = 2\mu\mathbf{S}, \tag{1.19}$$

where μ is the viscosity and $\mathbf{S}$ is the strain rate tensor, defined as the symmetric part of the velocity gradient as

$$\mathbf{S} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}}\right). \tag{1.20}$$

Inserting (1.18) and (1.19) to the Cauchy momentum equation (1.12) leads to the well-known incompressible Navier-Stokes equation

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{g}. \tag{1.21}$$

In (1.21) the Lagrangian derivative has been expanded in terms of the Eulerian derivative. It has also been used that for incompressible flow, assuming constant viscosity μ, since $\nabla \cdot \mathbf{u} = 0$

$$\nabla \cdot 2\mu\mathbf{S} = \mu\nabla^2\mathbf{u}. \tag{1.22}$$

The stress tensor of a viscoelastic fluid can be constructed through addition of an extra term to the deviatoric stress, such that

$$\boldsymbol{\tau}_d = 2\mu\mathbf{S} + \boldsymbol{\tau}, \tag{1.23}$$

where $\boldsymbol{\tau}$ is the viscoelastic stress. This leads to the momentum equation

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \nabla\cdot\boldsymbol{\tau} + \mathbf{g}, \tag{1.24}$$

which is simply the incompressible Navier-Stokes equation (1.21) with the additional term $\nabla \cdot \boldsymbol{\tau}$. The Newtonian contribution to viscosity μ may be either zero or nonzero depending on the type of viscoelastic fluid. In the context of polymeric fluids μ is often referred to as the solvent viscosity.

A constitutive model for the extra stress $\boldsymbol{\tau}$ is required to close the set of equations for the viscoelastic flow. In general, multiple relaxation modes can be used to model the viscoelastic stress as the sum of the individual stress modes, such that

$$\boldsymbol{\tau} = \sum_{k}^{N_m} \boldsymbol{\tau}_k. \tag{1.25}$$

In (1.25) $\boldsymbol{\tau}_k$ is the stress corresponding to the kth mode and N_m is the number of modes. Each mode is described by a constitutive equation with a unique set of parameters.

1.4.3 Convected derivatives

To construct a constitutive equation for the viscoelastic stress tensor it may appear feasible to, for example, replace the stress σ and strain rate $\dot{\gamma}$ in the Maxwell model equation (1.6) respectively by the second order tensors $\boldsymbol{\tau}$ and $2\mathbf{S}$. However, it can be shown that the resulting equation is not frame invariant (Morozov & Spagnolie, 2015). It is commonly accepted that for a constitutive equation to be admissible, it must be independent of the frame of reference in which it is expressed, including time-depending frames (Barnes *et al.*, 1989). This is sometimes expressed in terms of a material objectivity condition, see for example Lodge (1974).

To derive formulation principles for admissible constitutive equations, Oldroyd (1950) introduced a convected coordinate system $\{\hat{\mathbf{g}}_i\}_{i=1}^{d}$, where d is the spatial dimension, which is embedded in and deforms with the material. The vectors $\{\hat{\mathbf{g}}_i\}_{i=1}^{d}$ are often called the covariant base vectors. Another set of convected base vectors are the contravariant base vectors $\{\hat{\mathbf{h}}_i\}_{i=1}^{d}$, which are perpendicular to the material surfaces (Bird *et al.*, 1987b). Frame-invariant convected time derivatives can then be identified by expressing the stress tensor $\boldsymbol{\tau}$ in terms of the respective convected base vectors and taking the time derivative. After some manipulation, the covariant frame leads to the upper-convected derivative, which for a second order tensor $\mathbf{L}$ reads

$$\overset{\triangledown}{\mathbf{L}} = \frac{\mathrm{D}\mathbf{L}}{\mathrm{D}t} - \mathbf{L}\cdot\nabla\mathbf{u} - \nabla\mathbf{u}^{\mathsf{T}}\cdot\mathbf{L}, \tag{1.26}$$

and which appears in many common viscoelastic constitutive models. The contravariant frame leads to the lower-convected derivative,

$$\overset{\triangle}{\mathbf{L}} = \frac{\mathrm{D}\mathbf{L}}{\mathrm{D}t} + \mathbf{L}\cdot\nabla\mathbf{u}^{\mathsf{T}} + \nabla\mathbf{u}\cdot\mathbf{L}. \tag{1.27}$$

Furthermore, linear combinations of the upper-convected derivative (1.26) and the lower-convected derivative (1.27) also gives a frame-invariant time derivative as

$$\overset{\square}{\mathbf{L}} = \left(\frac{1+a_s}{2}\right)\overset{\triangledown}{\mathbf{L}} + \left(\frac{1-a_s}{2}\right)\overset{\triangle}{\mathbf{L}}, \tag{1.28}$$

where $a_s \in [-2,2]$ is called the slip parameter. Another way to express a general convected derivative is the Gordon-Schowalter derivative, which is often denoted $\overset{\square}{\mathbf{L}}$ and can be expressed as (Larson, 1988)

$$\overset{\square}{\mathbf{L}} = \frac{\mathrm{D}\mathbf{L}}{\mathrm{D}t} - \mathbf{L}\cdot\nabla\mathbf{u} - \nabla\mathbf{u}^{\mathsf{T}}\cdot\mathbf{L} + \xi\left(\mathbf{L}\cdot\mathbf{S} + \mathbf{S}\cdot\boldsymbol{\tau}\right), \tag{1.29}$$

where, $\xi \in [0,2]$ is related to the slip parameter a_s.

Higher order convected derivatives can be constructed by successive application of the convected derivative operator, such that (Bird *et al.*, 1987b)

$$\mathbf{L}_{[n+1]} = \left(\mathbf{L}_{[n]}\right)_{[1]}, \tag{1.30}$$

where the notation $\mathbf{L}_{[n]}$ refers to the convected derivative of order n and, thus, $\mathbf{L}_{[1]} = \overset{\square}{\mathbf{L}}$.

1.4.4 Normal stress differences

In Figure 1.3 an illustration of a simple shear flow is shown. For such a flow, the only nonzero velocity component is that in the x_1-direction, in this case denoted u_1. Furthermore, u_1 only varies in the x_2-direction. The velocity gradient tensor for the flow thus reads (Larson, 1999)

$$\nabla \mathbf{u} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 \\ \dot{\gamma} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{1.31}$$

where $\dot{\gamma} = \partial u_1 / \partial t$. For simple shear flow of a Newtonian fluid, the only nonzero component of the non-isotropic stress, i.e. the deviatoric stress $\boldsymbol{\tau}_d$, is the shear stress τ_{12}. If any other stress component is nonzero, the flow is by definition non-Newtonian (Barnes *et al.*, 1989).

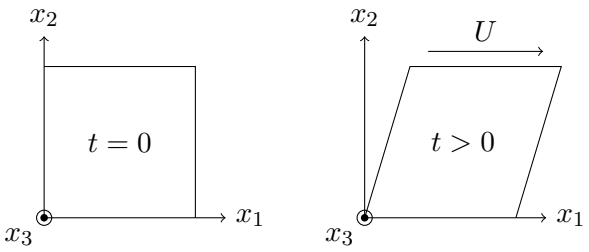

Figure 1.3: Simple shear flow.

In general, non-Newtonian fluids in simple shear flow exhibit nonzero normal components of $\boldsymbol{\tau}$. However, the stress tensor is only measurable up to an additive isotropic tensor. Therefore, only the normal stress differences $N_{1,2}$ can be measured (Larson, 1999), defined as

$$N_1 = \tau_{11} - \tau_{22}, \tag{1.32}$$

$$N_2 = \tau_{22} - \tau_{33}, \tag{1.33}$$

where N_1 and N_2 are called the first and second normal stress differences, respectively.

1.4.5 Constitutive models

A wide range of viscoelastic constitutive models can be found in the literature. In this section an overview is given. The primary focus is aimed to differential models, since the numerical method proposed in this thesis has been developed around this type of constitutive model.

Bird *et al.* (1987b) lists a number of ways in which admissible constitutive equations can be constructed. One such way is to generalize linear viscoelastic models by replacing the scalar quantities with their corresponding tensor counterparts and the time derivatives by an appropriate choice of a convected derivative. The result is so-called quasi-linear constitutive models. They also remark that only comparison with experimental data may provide insight to which form of the convected derivative operator that should be chosen.

Applying the above-mentioned generalization of linear viscoelasticity to (1.8) results in the equation

$$\sum_{i=0}^{n} \alpha_i \boldsymbol{\tau}_{[i]} = \beta_0' \boldsymbol{\gamma} + 2 \sum_{k=1}^{m} \beta_k \mathbf{S}_{[k-1]}, \tag{1.34}$$

in which the scalar stress σ has been replaced by the viscoelastic stress tensor $\boldsymbol{\tau}$ and the scalar strain γ by the strain tensor $\boldsymbol{\gamma}$. Furthermore, it can be shown (Bird *et al.*, 1987b) that $\boldsymbol{\gamma}_{[1]} = \overset{\square}{\boldsymbol{\gamma}} = \nabla \mathbf{u} + \nabla \mathbf{u}^{\mathsf{T}} = 2\mathbf{S}$. The nth order convected derivatives of $\boldsymbol{\gamma}$ has therefore been replaced by the $(n-1)$th order derivatives of $2\mathbf{S}$, which is commonly used in the context of viscoelastic fluid flow.

It was shown in Section 1.3 that the Maxwell element equation could be obtained by letting all coefficients be zero except for $\alpha_1 = \lambda$ and $\beta_1 = \eta$ in (1.8). Analogously, the same choice of parameters in (1.34) along with the upper-convected derivative, see (1.26), yields the upper-convected Maxwell (UCM) model. The UCM model has the constitutive equation

$$\lambda \overset{\triangledown}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2\eta \mathbf{S}, \tag{1.35}$$

where λ is the relaxation time and η the polymeric viscosity. Another famous constitutive model is the Oldroyd-B model, which can be obtained by letting $\alpha_1 = \lambda$, $\beta_0 = \eta_0$ and $\beta_1 = \eta_0 \lambda_r$ and has the constitutive equation (Larson, 1988)

$$\lambda \overset{\triangledown}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2\eta_0 \left(\mathbf{S} + \lambda_r \overset{\triangledown}{\mathbf{S}} \right), \tag{1.36}$$

where η_0 is the constant total viscosity and λ_r is called the retardation time. An alternative way to formulate the Oldroyd-B model is to split the deviatoric stress into a purely viscous part and a viscoelastic part as

$$\boldsymbol{\tau}_d = 2\frac{\lambda_r}{\lambda}\eta_0\mathbf{S} + \boldsymbol{\tau}, \tag{1.37}$$

where $\boldsymbol{\tau}$ has the constitutive equation

$$\lambda\overset{\triangledown}{\boldsymbol{\tau}} + \boldsymbol{\tau} = 2\left(1 - \frac{\lambda_r}{\lambda}\right)\eta_0\mathbf{S}. \tag{1.38}$$

In this form, the Oldroyd-B model is equivalent to the UCM model with nonzero solvent viscosity $\mu = \frac{\lambda_r}{\lambda}\eta_0$ and polymeric viscosity $\eta = \left(1 - \frac{\lambda_r}{\lambda}\right)\eta_0$.

Quasi-linear models, including the UCM and Oldroyd-B models, do not impose an upper limit as to how much polymer molecules may be physically stretched. They may therefore produce unbounded normal stress. A remedy to this is the use of nonlinear models, which can provide a more physically correct description for a wider range of flows.

Another way to construct admissible constitutive equations, as pointed out by Bird *et al.* (1987b), is to combine convected derivatives with empirical expressions. The Oldroyd-B model can be considered such a model. Another example is the Giesekus model, which is obtained by adding a quadratic term to the UCM model (Morozov & Spagnolie, 2015),

$$\lambda\overset{\triangledown}{\boldsymbol{\tau}} + \boldsymbol{\tau} + \frac{\alpha_G\lambda}{\eta}\boldsymbol{\tau}\cdot\boldsymbol{\tau} = 2\eta\mathbf{S}, \tag{1.39}$$

where $\alpha_G \in [0, 1/2]$ is a dimensionless parameter.

So far, the constitutive equations discussed have been constructed directly at the continuum level. Another approach listed by Bird *et al.* (1987a) is to use molecular theories. This involves assuming some kind of model for the macromolecules in the fluid and for their interaction. It also requires some closure model or averaging to procedure to reach a constitutive equation that describe the stresses at the continuum level.

One example of such a model is the Phan Thien Tanner (PTT) (Thien & Tanner, 1977). The model was derived by modeling polymer molecules in a network with non-affine motions between the network strands. The proposed constitutive equation reads

$$\lambda\overset{\square}{\boldsymbol{\tau}} + \left(1 + \frac{\varepsilon\lambda}{\eta}\mathrm{Tr}\left(\boldsymbol{\tau}\right)\right)\boldsymbol{\tau} = 2\eta\mathbf{S}, \tag{1.40}$$

where ε is a dimensionless parameter and $\mathrm{Tr}\,(\boldsymbol{\tau})$ is the trace of $\boldsymbol{\tau}$, i.e. the sum of its normal components such that $\mathrm{Tr}\,(\boldsymbol{\tau}) = \tau_{xx} + \tau_{yy} + \tau_{zz}$. A modified version of the constitutive equation with an exponential form of the nonlinear term was later proposed, reading

$$\lambda \overset{\square}{\boldsymbol{\tau}} + \exp\left(\frac{\varepsilon\lambda}{\eta}\mathrm{Tr}\,(\boldsymbol{\tau})\right)\boldsymbol{\tau} = 2\eta\mathbf{S}. \tag{1.41}$$

If the full Gordon-Schowalter derivative $\overset{\square}{\boldsymbol{\tau}}$ is replaced by the upper convected derivative $\overset{\triangledown}{\boldsymbol{\tau}}$ the model is sometimes referred to as the simplified PTT (SPTT), model.

Another family of models derived from molecular theory is the finitely extensible nonlinear elasticity (FENE) models (Herrchen & Öttinger, 1997). The FENE models are derived by treating the viscoelastic fluid as a dilute solution of nonlinear dumbbells, i.e. beads connected by a nonlinear spring. A sketch of a dumbbell is shown in Figure 1.4. A closure approximation is required to derive a constitutive equation for the continuum scale stress, and different approximations yield different constitutive equations.

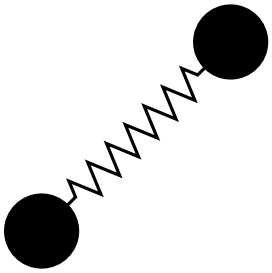

Figure 1.4: Sketch of a dumbbell.

One such closure model is the Peterlin approximation which yields the FENE-P constitutive equation (Bird *et al.*, 1987b)

$$Z(\mathrm{Tr}\,(\boldsymbol{\tau}))\boldsymbol{\tau} + \lambda\overset{\triangledown}{\boldsymbol{\tau}} - \lambda\left(\boldsymbol{\tau} - (1 - \frac{2}{2+l})\frac{\eta}{\lambda}\mathbf{I}\right)\frac{\mathrm{D}\ln Z}{\mathrm{D}t} = -2(1 - \frac{b}{b+2})\eta\mathbf{S}, \tag{1.42}$$

where l is a model parameter related to the maximum dumbbell extension and the function Z reads

$$Z(\mathrm{Tr}\,(\boldsymbol{\tau})) = 1 + \frac{3}{l}\left(1 - \frac{\lambda}{3\eta}\mathrm{Tr}\,(\boldsymbol{\tau})\right). \tag{1.43}$$

Another closure approximation was proposed by Chilcott & Rallison (1988), yielding the FENE-CR model. The constitutive can be written as

$$\lambda \overset{\triangledown}{\boldsymbol{\tau}} + \left(\frac{l^2 + \frac{\lambda}{\eta}\mathrm{Tr}\left(\boldsymbol{\tau}\right)}{l^2 - 3} \right) \boldsymbol{\tau} = 2\eta \left(\frac{l^2 + \frac{\lambda}{\eta}\mathrm{Tr}\left(\boldsymbol{\tau}\right)}{l^2 - 3} \right) \mathbf{S}, \tag{1.44}$$

where, again, l is related to the maximum dumbbell extension.

1.4.6 Conformation tensor

In addition to stress the tensor $\boldsymbol{\tau}$, the viscoelastic state may also be expressed in terms of the molecular configuration state of the material through the dimensionless conformation tensor $\mathbf{c}$. The conformation tensor is related to the stress tensor as

$$\mathbf{c} = \frac{\lambda}{\eta}\boldsymbol{\tau} + \mathbf{I}. \tag{1.45}$$

Assuming that the viscoelastic fluid consists of some type of polymer chains, $\mathbf{c}$ is the second moment of the dimensionless end-to-end vectors of polymer chains (Morozov & Spagnolie, 2015). Many constitutive models can be expressed in terms of $\mathbf{c}$ in the general form (Chen *et al.*, 2013)

$$\overset{\triangledown}{\mathbf{c}} = \frac{1}{\lambda}\mathcal{Y}(\mathbf{c})\boldsymbol{\mathcal{H}}(\mathbf{c}), \tag{1.46}$$

where $\mathcal{Y}(\mathbf{c})$ is a scalar-valued function and $\boldsymbol{\mathcal{H}}(\mathbf{c})$ a tensor-valued function. The functions $\mathcal{Y}$ and $\boldsymbol{\mathcal{H}}$ for the constitutive models discussed in the previous section are listed in Table 1.1.

Model	$\mathcal{Y}(\mathbf{c})$	$\boldsymbol{\mathcal{H}}(\mathbf{c})$
UCM/Oldroyd-B	1	$\mathbf{I} - \mathbf{c}$
Giesekus	1	$\mathbf{I} - \mathbf{c} - \alpha_G(\mathbf{I} - \mathbf{c})$
(S)PTT (linear form)	$1 + \varepsilon(\mathrm{Tr}\left(\mathbf{c}\right) - 3)$	$\mathbf{I} - \mathbf{c}$
(S)PTT (exponential form)	$\varepsilon(\mathrm{Tr}\left(\mathbf{c} - 3\right)$	$\mathbf{I} - \mathbf{c}$
FENE-P	1	$\mathbf{I} - \mathbf{c}/(1 - \mathrm{Tr}\left(\mathbf{c}\right)/l^2)$
FENE-CR	$(1 - \mathrm{Tr}\left(\mathbf{c}\right)/l^2)^{-1}$	$\mathbf{I} - \mathbf{c}$

Table 1.1: Functions $f(\mathbf{c})$ and $\boldsymbol{\mathcal{H}}(\mathbf{c})$ in (1.46) for various constitutive models.

The viscoelastic stress can be calculated directly by solving (1.46) for the conformation tensor. Furthermore, the conformation tensor is commonly used in different types of stability enhancement techniques. This is discussed in detail in Section 1.5.2.

1.4.7 Other models

In addition to the models discussed above, various constitutive equations may be found in the literature. Some additional examples of differential models derived using molecular theory are the Pom-pom model (McLeish & Larson, 1998) for branched polymer melts and the Rolie-Poly model (Likhtman & Graham, 2003). Another class of constitutive equation are integral models, where the influence of the deformation history is described in terms of a memory function. One such example is the K-BKZ model, see for example Mitsoulis (2013). A detailed discussion about these models is outside the scope of this thesis.

1.5 A review of numerical methods for viscoelastic flow

Different aspects on numerical simulations of viscoelastic flows are discussed in this section, based on the literature. The intent is to provide the reader with an overview and not to give a complete review of the field.

A common approach for simulation of viscoelastic flow is to solve all equations in the Eulerian frame of reference using the finite volume or finite element method. For some examples of finite volumes see Alves *et al.* (2001, 2003); Oliveira *et al.* (1998), for some example of finite elements, see Baaijens *et al.* (1995); Hulsen *et al.* (2005). The Eulerian frame of reference is suitable for diffusion-dominated applications, such as viscous flow or heat and mass transfer. Viscoelastic constitutive equations are however hyperbolic and specialized high-order discretization schemes may be required for the convective term, in order to minimize numerical diffusion (Alves *et al.*, 2003).

An alternative approach is to solve all or a subset of the involved equations in the Lagrangian frame of reference. Rasmussen & Hassager (1995) developed a Lagrangian method to solve the equations for the flow of an UCM fluid with a finite element discretization. The entire deformation history of the flow was stored, and re-meshing was required throughout the simulation. Harlen *et al.* (1995) proposed a split Lagrangian-Eulerian method, in which viscoelastic Stokes flow was simulated by solving the constitutive equation at the nodes of a co-deforming mesh. The velocity and pressure fields obtained by solving the momentum and continuity equations using a Eulerian finite element method. Delaunay triangulation was utilized to reconnect the nodes as the mesh became distorted. Halin *et al.* (1998) proposed a finite element

method in which the constitutive equation was integrated along particle trajectories. The method was denoted the Lagrangian particle method (LPM). Local polynomial approximations were then fitted to the stress in each element, such that the local finite element integrals could be evaluated. Thus, at least three particles were required in each two-dimensional element in order for the simulations not to fail, since three data points are required to fit a first order polynomial. Their method was later refined to the adaptive Lagrangian particle method (ALPM) (Gallez *et al.*, 1999), in which particles were adaptively created and deleted. For both methods, a relatively large number of particles was required for stable transient results. A backward-tracking version of the method was later proposed, denoted the backward-tracking Lagrangian particle method (BLPM) (Wapperom *et al.*, 2000).

1.5.1 The Reynolds, Weissenberg and Deborah numbers

Three dimensionless number are of importance for quantifying the characteristics of viscoelastic flows. The Reynolds number quantifies the ratio of inertial to frictional forces in the flow, and reads (Schlichting & Gersten, 2000)

$$\mathrm{Re} = \frac{\rho U L}{\mu}, \tag{1.47}$$

where U and L are characteristic velocity and length scales, respectively. The Reynolds number is frequently used in computational fluid dynamics for Newtonian as well as non-Newtonian flows. While for Newtonian fluids the viscosity is a constant, for some non-Newtonian fluids it is not. A characteristic viscosity then needs to be chosen.

Two additional dimensionless numbers that describe the effects of elasticity in viscoelastic flows are the Deborah number and the Weissenberg number. It is remarked that, while they for many flows are similarly defined, they have different physical meaning. The Deborah number was originally proposed by Reiner (1964) as

$$\mathrm{De} = \frac{\text{Characteristic material time scale}}{\text{Observation time scale}}. \tag{1.48}$$

If the relaxation time is short in relation to the observation time, i.e. if De is small, fluid-like behavior is expected. If the opposite is true, i.e. if De is large, solid-like behavior is expected. For practical reasons, the now commonly used definition is

$$\mathrm{De} = \frac{t_{\mathrm{relaxation}}}{t_{\mathrm{process}}}, \tag{1.49}$$

where $t_{\text{relaxation}}$ is the relaxation time of the material and t_{process} the time scale of the deformation process. It is remarked that for certain flows that are steady in the Lagrangian sense, such that fully developed pipe and channel flows or steady simple shear, the time scale of the deformation time is infinite. Hence, the Deborah number is zero for such flows (Poole, 2012).

The Weissenberg number Wi was identified by White (1964) through dimensional analysis of the equations of motion for steady flow of a second order fluid. Following his procedure, the second order fluid has the extra stress tensor

$$\boldsymbol{\tau}_d = \mu \mathbf{B}_1 + \omega_2 \mathbf{B}_1^2 + \omega_3 \mathbf{B}_2, \tag{1.50}$$

where $\mathbf{B}_1 = 2\mathbf{S}$, $\mathbf{B}_2 = \frac{\mathrm{D}}{\mathrm{D}t}\mathbf{B}_1 - \nabla\mathbf{u}\cdot\mathbf{B}_1 - \mathbf{B}_1\cdot\nabla\mathbf{u}$, and $\omega_{2,3}$ are coefficients. The incompressible momentum equation for the second order fluid can thus be expressed, for steady flow and with the body force term omitted, as

$$\rho\mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \mu\nabla^2\mathbf{u} + \omega_2\nabla\cdot\mathbf{B}_1^2 + \omega_3\nabla\cdot\mathbf{B}_2. \tag{1.51}$$

The equation (1.51) can be expressed in terms of dimensionless variables as

$$\hat{\mathbf{u}}\cdot\hat{\nabla}\hat{\mathbf{u}} = -\hat{\nabla}\left(\frac{p}{\rho U^2}\right) + \frac{1}{\mathrm{Re}}\hat{\nabla}^2\hat{\mathbf{u}} - \frac{\mathrm{Wi}}{\mathrm{Re}}\left[\hat{\nabla}\cdot\hat{\mathbf{B}}_2 + \left(\frac{\omega_2}{\omega_3}\right)\hat{\nabla}\cdot\hat{\mathbf{B}}_1^2\right], \tag{1.52}$$

where $\hat{(\bullet)}$ denotes dimensionless variables. In (1.52) Re is the Reynolds number as defined in (1.47) and the group

$$\mathrm{Wi} = -\frac{\omega_3 U}{\mu L} = J_e\mu\frac{U}{L} \tag{1.53}$$

represents the ratio of viscoelastic to viscous forces and was named the Weissenberg number. The minus sign in (1.53) comes from the observation that $\mu > 0$ and $\omega_3 < 0$ in laminar shear flow and J_e is the steady state shear compliance. The factor $J_e\mu$ has the dimension of time and the characteristic material relaxation time λ is often used to calculate Wi.

Following Poole (2012), for steady simple shear flow of an UCM fluid, the elastic forces are characterized by the first normal stress difference $N_1 = \tau_{xx} - \tau_{yy} = 2\lambda\eta\dot{\gamma}^2$ and the viscous forces by the shear stress $\tau_{xy} = \eta\dot{\gamma}$, where $\dot{\gamma}$ is a characteristic shear rate. The Weissenberg number is then

$$\mathrm{Wi} = \frac{N_1}{\tau_{xy}} = \frac{2\lambda\eta\dot{\gamma}^2}{\eta\dot{\gamma}} = 2\lambda\dot{\gamma} = 2\lambda\frac{U}{L}. \tag{1.54}$$

From (1.54) it is clear that Wi can be interpreted as a ratio of viscoelastic to viscous forces. It is remarked that in some cases the Deborah and Weissenberg numbers are equal or are related through a geometrical factor. However, as pointed out by Poole (2012), this not is the case for all flows.

Finally, it is noted that the so-called elasticity number, defined as (Owens & Phillips, 2002)

$$\mathrm{El} = \frac{\mathrm{Wi}}{\mathrm{Re}}, \tag{1.55}$$

is useful to quantify the ratio of elastic effects and inertial effects in viscoelastic flow.

1.5.2 Numerical stability considerations

An important aspect of viscoelastic flow simulations is numerical stability. Instabilities and convergence issues may arise in numerical simulations already for moderately large Weissenberg or Deborah numbers. This is commonly referred to as the high Weissenberg number problem (HWNP) (Keunings, 2000).

Convergence issues at limiting values of Wi or De are either due to model limitations or numerical approximation errors. While some semi-analytical evidence of limiting values exist, in fact what can be interpreted as a limit for the discretized problem is often numerical artifacts (Owens & Phillips, 2002). Numerical evidence has suggested that the numerical breakdown is due to a loss of resolution near large stress gradients, and that commonly used polynomial approximations are inappropriate to represent the stress profiles in regions where the stress growth is exponential (Fattal & Kupferman, 2004).

Various approaches to remedy the HWNP have been suggested. It is however remarked that, as pointed out by Keunings (2000), introducing schemes or modifications that either implicitly or explicitly smooths out the stress profiles may provide numerical stability, but may also change the problem being solved to a different than the original one.

Different methods have been proposed to remedy the HWNP. One approach is to design methods that preserve the positive definiteness of the conformation tensor. A few examples of such methods are given below. The stabilization methods presented in this section are commonly applied to Eulerian solution algorithms, using either finite volume or finite element discretization.

A fairly simple method to increase numerical stability of viscoelastic flow simulations is to enhance the ellipticity of the problem by adding artificial diffusion on both sides of the momentum equation. This is known as both sides diffusion (BSD). The momentum equation (1.24) can then be expressed as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\nabla\cdot\mathbf{u}\right)-2(\mu+\mu_a)\nabla\cdot\mathbf{S}=-\nabla p-2\mu_a\nabla\cdot\mathbf{S}+\nabla\cdot\boldsymbol{\tau}+\mathbf{g}, \tag{1.56}$$

where μ_a is an artificial viscosity. In the continuum sense, the terms $2\mu_a\nabla\cdot\mathbf{S}$ on both sides cancel each other. However, when discretizing the equations, the term on the left hand side is treated implicitly and the one on the right hand side is treated explicitly. Consequently, a small amount of numerical diffusion is introduced, which enhances the numerical stability. While the method can be suitable for calculation of steady flow, it can falsely diffuse the solution in time and should preferably be avoided for transient simulations. This issue is for example pointed out by Xue *et al.* (2004).

A popular stabilization approach is the log-conformation representation (LCR), proposed by Fattal & Kupferman (2004, 2005). The main idea of the LCR is to reduce the steep exponential growth of the viscoelastic stress in certain regions of the flow. This is achieved by reformulating the constitutive equation to an equation for the matrix logarithm of the conformation tensor.

For a divergence-free velocity field, i.e. $\nabla\cdot\mathbf{u}=0$, the velocity gradient may be locally decomposed as

$$\nabla\mathbf{u}=\boldsymbol{\Omega}+\mathbf{B}+\mathbf{N}\mathbf{c}^{-1}, \tag{1.57}$$

where $\boldsymbol{\Omega}$ and $\mathbf{N}$ are antisymmetric tensors, i.e. pure rotations, $\mathbf{B}$, commutes with $\mathbf{c}$ and $\mathrm{Tr}\,(\mathbf{B})=0$. Then for a constitutive equation on the form (1.46), $\boldsymbol{\Theta}=\log(\mathbf{c})$ satisfies the equation

$$\frac{\mathrm{D}\boldsymbol{\Theta}}{\mathrm{D}t}-(\boldsymbol{\Omega}\boldsymbol{\Theta}-\boldsymbol{\Theta}\boldsymbol{\Omega})-2\mathbf{B}=\frac{1}{\lambda}\mathcal{Y}(e^{\boldsymbol{\Theta}})e^{-\boldsymbol{\Theta}}\mathcal{H}(e^{\boldsymbol{\Theta}}). \tag{1.58}$$

A different stabilization approach, which by construction maintains symmetry of the conformation tensor, is the square-root conformation representation (SRCR) by Balci *et al.* (2011). The square-root conformation tensor $\mathbf{b}$ is defined such that

$$\mathbf{c}=\mathbf{b}\cdot\mathbf{b}, \tag{1.59}$$

i.e. its square is the conformation tensor. In their original work, Balci *et al.* (2011) derived the equation for $\mathbf{b}$ for the Oldroyd-B and FENE-P models. For a general

constitutive equation on the form (1.46), the equation for $\mathbf{b}$ reads (Palhares Junior *et al.*, 2016)

$$\frac{\mathrm{D}\mathbf{b}}{\mathrm{D}t} = \mathbf{b}\cdot\nabla\mathbf{u}^{\mathsf{T}} + \mathbf{M}\cdot\mathbf{b} + \frac{1}{2\lambda}\mathcal{Y}(\mathbf{b}^2)\mathbf{b}^{-1}\cdot\mathcal{H}(\mathbf{b}^2), \tag{1.60}$$

where $\mathbf{M}$ is antisymmetric. The components of $\mathbf{M}$ can be calculated explicitly by observing that the tensor $\mathbf{b}\cdot\nabla\mathbf{u}^{\mathsf{T}} + \mathbf{M}\cdot\mathbf{b}$ must be symmetric for (1.60) to preserve the symmetry of $\mathbf{b}$.

Another approach is the positive definiteness preserving scheme (PDPS) proposed by Stewart *et al.* (2008). The idea behind the approach was to design a discretization scheme for the constitutive equation which preserves the positive definiteness of the conformation tensor by construction.

The discussed stabilization methods have different properties in terms of stability, accuracy and implementation complexity. For a thorough comparison of the methods, see Chen *et al.* (2013). They concluded that since BSD does not preserve positive definiteness of $\mathbf{c}$, it is insufficient to fully solve the HWNP. However, one advantage of BSD is that it can solve instabilities caused by low viscosity ratios, i.e. small solvent viscosity compared to polymeric viscosity. BSD also has low implementation complexity and may be combined with other stabilization methods. Furthermore, they found that LCR, SRCR and PDPS are stable at higher Weissenberg numbers, and that LCR had higher convergence rate than the other two. However, they state that at least for moderate Weissenberg numbers, SRCR is an optimal choice. This is mainly attributed to relatively low implementation complexity and computational cost.

1.6 Motivation for the choice of method

Eulerian finite volume discretization is a well-established approach for solving problems where diffusion is important, including viscous flow and transport of heat and mass. However, many viscoelastic constitutive equations do not have a physical diffusion term. The Lagrangian frame of reference is therefore in a sense a natural frame in which to solve the constitutive equation. A combination of the Lagrangian and Eulerian frames, respectively, is therefore suitable.

Another motivation is that the Lagrangian formulation brings certain appealing properties in terms of computational efficiency. A major part of the proposed Lagrangian-Eulerian method to calculate the viscoelastic stress is straightforward to parallelize (In-

gelsten *et al.*, 2019). As will be shown in this thesis, it is therefore suitable for GPU-acceleration. This should be viewed in contrast to solving three or six coupled matrix equations, respectively for 2D and 3D, which is required for a Eulerian finite volume discretization of the constitutive equation. Furthermore, the method is suitable for multiphase flows, since the Lagrangian fluid nodes only needs to be generated in the part of the domain occupied by the viscoelastic fluid phase.

1.7 Software framework

The methodology presented in this thesis is implemented in the in-house software platform IPS IBOFlow® (IPS IBOFlow, 2020), an incompressible fluid flow solver developed at the Fraunhofer-Chalmers Centre for Industrial Mathematics in Gothenburg, Sweden. The key features of the numerical framework are the use of the mirroring immersed boundary method (Mark & van Wachem, 2008; Mark *et al.*, 2011) and the use of an automatically generated and adaptively refined Cartesian octree mesh. Prior to the current work, the software has been employed for simulation of conjugated heat transfer (Mark *et al.*, 2013) and fluid-structure interaction (Svenning *et al.*, 2014). Furthermore, the solver has been used for simulation of two-phase flows of shear-thinning fluids for seam sealing (Edelvik *et al.*, 2017; Mark *et al.*, 2014), adhesive application (Svensson *et al.*, 2016) and 3D-bioprinting (Göhl *et al.*, 2018).

Chapter 2

Numerical method

In this chapter the proposed numerical method is presented. The two main components of the numerical framework are a Eulerian finite volume method used to solve the momentum and continuity equations and a Lagrangian solver for the viscoelastic constitutive equation. The solvers have different discretization and frames of reference. Hence, the coupling between the fields through interpolation is also an important component of the algorithm.

The main scientific contribution of this work is the Lagrangian framework to solve the viscoelastic stresses and the coupling to the Eulerian finite volume solver through unstructured interpolation. The Lagrangian formulation of the viscoelastic constitutive equation allows for it to be solved efficiently using parallel computation on the CPU or the GPU. This enables fast simulations, which is a requirement for many industrial applications. The CPU and GPU implementations of the method are discussed in Section 2.4.

2.1 Governing equations

The viscoelastic fluid flow is described by the incompressible momentum and continuity equations along with a constitutive equation for the viscoelastic stress. The momentum equation (1.24), including BSD with artificial viscosity μ_a, reads

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\nabla\cdot\mathbf{u}\right) - \nabla\cdot(2(\mu+\mu_a)\mathbf{S}) = -\nabla p + \nabla\cdot\boldsymbol{\tau} - \nabla\cdot(2\mu_a\mathbf{S}) + \nabla\cdot\boldsymbol{\tau} + \mathbf{g}, \quad (2.1)$$

and the continuity equation, included again here for completeness, reads

$$\nabla\cdot\mathbf{u} = 0. \quad (1.11)$$

The viscoelastic stress is assumed to have N_m relaxation modes, with each modal stress described by a constitutive equation of the general form

$$\lambda_k \overset{\square}{\boldsymbol{\tau}}_k = \boldsymbol{\mathcal{F}}_k(\boldsymbol{\tau}_k, \nabla \mathbf{u}), \tag{2.2}$$

where $\boldsymbol{\mathcal{F}}_k$ is a tensor-valued function which is defined by the choice of constitutive model. Thus, by expanding the Gordon-Schowalter derivative in (2.2), the stress mode $\boldsymbol{\tau}_k$ in a fluid element is described by the Lagrangian time derivative

$$\frac{\mathrm{D}\boldsymbol{\tau}_k}{\mathrm{D}t} = \frac{1}{\lambda_k}\boldsymbol{\mathcal{F}}_k(\boldsymbol{\tau}_k, \nabla \mathbf{u}) + \mathbf{A}(\boldsymbol{\tau}_k), \tag{2.3}$$

where

$$\mathbf{A}(\boldsymbol{\tau}_k) = \boldsymbol{\tau}_k \cdot \nabla \mathbf{u} + \nabla \mathbf{u}^T \cdot \boldsymbol{\tau}_k - \xi \left(\boldsymbol{\tau}_k \cdot \mathbf{S} + \mathbf{S} \cdot \boldsymbol{\tau}_k \right). \tag{2.4}$$

2.2 Eulerian finite volume solver

The momentum and continuity equations are discretized and solved with the finite volume method and on a collocated Cartesian octree grid. The grid is automatically generated and can be adaptively refined, for example if objects in the domain are moving.

The momentum equation (1.24) gives a transport equation for each velocity component and is coupled to the pressure field through the pressure gradient. However, the incompressible continuity equation (1.11) does not include the pressure. A pressure-velocity coupling algorithm is therefore required and there exist several algorithms in the literature. In this work the Semi-Implicit Method for Pressure-Linked Equations Consistent (SIMPLEC) (Doormaal & Raithby, 1984), based on the original SIMPLE method by Patankar (1980), is used.

2.2.1 Immersed boundary method

Interior objects in the computational domain are represented by surface triangulations and boundary conditions from the objects are imposed using the mirroring immersed boundary method (Mark & van Wachem, 2008; Mark *et al.*, 2011). In the method, the velocity field is implicitly mirrored across the boundary surface when the momentum matrix is assembled, such that the prescribed boundary condition is satisfied for the converged solution. The mirroring introduces a fictitious flow field inside the object. To

ensure zero mass flow across the boundary, these velocities are replaced by the object velocity in all flux calculations.

2.2.2 Volume of fluid method

Multiphase flow is modeled with the volume of fluid (VOF) method. The local amount of each phase is described in terms of the volume fraction $\alpha \in [0, 1]$. The transport of the volume fraction is described by the equation (Tryggvason *et al.*, 2011)

$$\frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha = 0, \tag{2.5}$$

which is solved on the Eulerian grid. The volume of each phase in a cell with volume ΔV is $\alpha \Delta V$ and $(1 - \alpha)\Delta V$, respectively. In a cell where $\alpha = 1$, only phase one is present. Conversely, if $\alpha = 0$ only phase two is present. If $0 < \alpha < 1$ the cell lies at the interface between the two phases. To avoid smearing the interface between the phases through numerical diffusion, the convection term in (2.5) is discretized with the compact CICSAM scheme (Ubbink & Issa, 1999).

A single set of momentum and continuity equations is solved for whole computational domain. Local properties, e.g. density or viscosity, are calculated as the average

$$\phi = \alpha \phi_1 + (1 - \alpha)\phi_2, \tag{2.6}$$

where ϕ_1 is the property of first phase and ϕ_2 that of the second phase .

2.3 Lagrangian-Eulerian viscoelastic stress solver

The constitutive equation (2.2) is solved in the Lagrangian frame of reference, along the trajectories of Lagrangian nodes, or fluid elements, convected by the flow. The formulation results in a coupled ODE system for the node position and stress. The stresses are then interpolated from the unstructured Lagrangian node set to the Eulerian grid using radial basis functions (Iske, 2004). The stress contribution is added to the discretized momentum equation by integrating the divergence of the viscoelastic stress in (2.1). The procedure carried out in each time step can be summarized as:

1. Distribute/redistribute Lagrangian nodes

2. Solve ODE systems

3. Interpolate viscoelastic stresses to Eulerian fluid grid

4. Calculate pseudo-force from the stresses and add to the momentum equation

The different steps are discussed in detail below.

2.3.1 Distribution of Lagrangian nodes

At initialization, the Lagrangian nodes are distributed throughout the computational domain as follows: A subdivision is obtained by splitting each cell into n_{split} smaller segments in each spatial direction. Consequently, a cell holds n_{split}^d sub-volumes, each in which a Lagrangian node is created at its center. The division and initial distribution in a two-dimensional cell are shown in Figure 2.1 for $n_{\text{split}} = 2$ and $n_{\text{split}} = 3$.

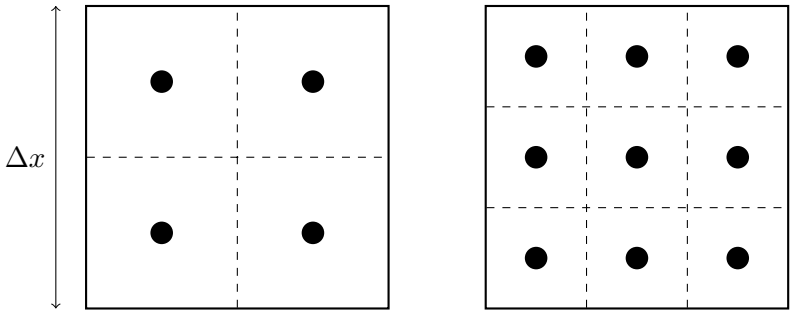

Figure 2.1: Subdivision of a two-dimensional Eulerian cell for distribution of Lagrangian nodes with $n_{\text{split}} = 2$ (left) and $n_{\text{split}} = 3$ (right).

Since the Lagrangian nodes are not stationary, the distribution is maintained through addition and deletion once per global time step. A node is added if the neighborhood of a sub-volume is empty. The neighborhood is defined as the box sharing its center with the sub-volume and with the side $\Delta x/n_{\text{split}}(1 + \varepsilon_{\text{neigh}})$, with $\varepsilon_{\text{neigh}} > 0$. This is illustrated in Figure 2.2. In this work $\varepsilon_{\text{neigh}} = 0.1$ has been found suitable and is therefore used. If a node is added, its stress is interpolated from the surrounding nodes in the current node set with the RBF method, i.e. using (2.12).

A number $n_{\max} \geq 1$ is defined that denotes the maximum allowed number of nodes in a sub-volume. If the number of nodes in a sub-volume exceeds $n_{\max}$, nodes are deleted until the number is within the acceptable range. The deletion is carried out

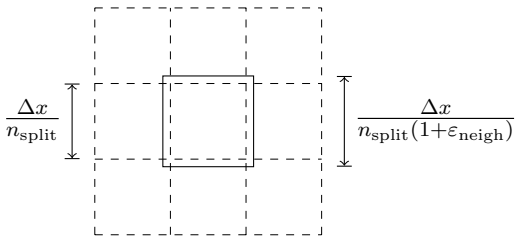

Figure 2.2: Box covering subcell.

by identifying the pairwise closest nodes, delete them and create a new node at their middle position and initialized with their mean stress.

The strategy for addition and deletion is partly inspired by Gallez *et al.* (1999). A difference is that their deletion step was performed for the whole cell, while in this work it is performed for each sub-cell individually.

2.3.2 ODE systems

Fluid elements in the flow field are described as mass-less particles, or nodes. The position $\mathbf{x}$ of a node is described by the ODE

$$\dot{\mathbf{x}} = \mathbf{u}, \tag{2.7}$$

The stress modes $\{\boldsymbol{\tau}_k\}_{k=1}^N$ are given by (2.3). An ODE system for the position and stress in a node resulting from (2.7) and (2.3) then reads

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{u} \\ \dot{\boldsymbol{\tau}}_1 &= G_1(\boldsymbol{\tau}_1, \nabla\mathbf{u}) \\ &\vdots \\ \dot{\boldsymbol{\tau}}_N &= G_N(\boldsymbol{\tau}_N, \nabla\mathbf{u}) \end{cases}, \tag{2.8}$$

where $G_k(\boldsymbol{\tau}_k, \nabla\mathbf{u})$ follows from (2.3). When $\mathbf{u}$ and $\nabla\mathbf{u}$ are required at the current position of a node, they are interpolated from the Eulerian grid using bilinear or trilinear interpolation, respectively for 2D and 3D simulations. The concept of a Lagrangian node traveling along its trajectory in the flow field is visualized in Figure 2.3.

In this work, two algorithms to solve the system (2.8) are considered, a second order backward differentiation formula (BDF) and the implicit Euler method. However, it is remarked that, in principle, any suitable ODE solver may be chosen.

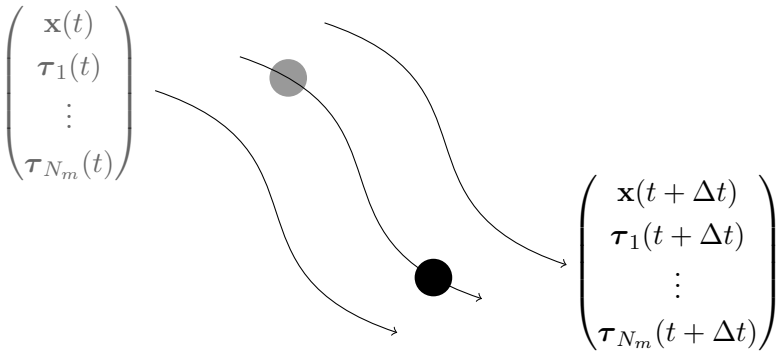

Figure 2.3: Concept of Lagrangian node trajectory in the fluid flow field.

Let $\mathbf{y}$ be the ODE solution vector

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} & \boldsymbol{\tau}_1 & \cdots & \boldsymbol{\tau}_N \end{bmatrix}^{\mathsf{T}}, \tag{2.9}$$

and let Δt denote the global step size, i.e. the time step used to solve the momentum and continuity equations. The ODE systems are solved from time t to $t + \Delta t$ using $N_{\text{loc}} > 0$ local steps with lengths Δt_n, $n = 1, \ldots, N_{\text{loc}}$, such that $\sum_n \Delta t_n = \Delta t$.

For the implicit Euler method, an approximate solution at time t_n is calculated from

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \Delta t_n \dot{\mathbf{y}}_n, \tag{2.10}$$

where subscript n denotes a property at time t_n and $\Delta t_n = \Delta t / N_{\text{loc}}$. For the BDF method, an approximate solution $\mathbf{y}_n$ at time t_n is calculated from

$$b_n \Delta t_n \dot{\mathbf{y}}_n - \mathbf{y}_n + a_{n,1}\mathbf{y}_{n-1} - a_{n,2}\mathbf{y}_{n-2} = \mathbf{0}, \tag{2.11}$$

where $\Delta t_n = t_n - t_{n-1}$ is of variable length and the constants b_n, $a_{n,1}$ and $a_{n,2}$ are uniquely determined given the recent step size history (Hindmarsh *et al.*).

2.3.3 Unstructured interpolation

Radial basis functions (RBF) are used to interpolate the viscoelastic stress from the Lagrangian nodes to the Eulerian grid. Consider a function $f : \mathbb{R}^d \mapsto \mathbb{R}$, where d is the spatial dimension, for which the values are known in the points $\{\mathbf{x}_n\}_{n=1}^{N_c}$, $N_c > 0$. The

interpolant of f at a point $\mathbf{r}$ is then calculated as (Iske, 2004)

$$\hat{f}(\mathbf{r}) = \sum_{n=1}^{N_c} w_n \psi(\zeta_s |\mathbf{r} - \mathbf{x}_n|) + P(\mathbf{r}), \tag{2.12}$$

where $\{w_n\}_{n=1}^{N_c}$ are weights, $\psi : \mathbb{R} \mapsto \mathbb{R}$ a radial basis function and ζ_s a scaling parameter. The second term on the right hand side of (2.12) P is the polynomial

$$P(\mathbf{r}) = v_0 + \sum_{s=0}^{d} v_s r_s, \tag{2.13}$$

where $\{v_s\}_{s=0}^{d}$ are polynomial coefficients and $\{r_s\}_{s=1}^{d}$ the components of $\mathbf{r}$ in the respective spatial directions. The coefficients in (2.12) and (2.13) satisfy the system

$$\begin{bmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \tag{2.14}$$

where

$$M_{ij} = \psi(\zeta_s |\mathbf{x}_i - \mathbf{x}_j|), \tag{2.15}$$

$$\mathbf{B} = \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{x}_1 & \cdots & \mathbf{x}_{N_c} \end{bmatrix}^T, \tag{2.16}$$

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & \cdots & f(\mathbf{x}_{N_c}) \end{bmatrix}, \tag{2.17}$$

$$\mathbf{w} = \begin{bmatrix} w_1 & \cdots & w_{N_c} \end{bmatrix}, \tag{2.18}$$

$$\mathbf{v} = \begin{bmatrix} v_0 & \cdots & v_d \end{bmatrix}. \tag{2.19}$$

It is remarked that the matrices $\mathbf{M}$ and $\mathbf{B}$ depend only the distances between the points while the vector $\mathbf{f}$, and thus also $\mathbf{w}$ and $\mathbf{v}$, depend on f.

The interpolation of the viscoelastic stress to a point is carried out as follows: The Lagrangian nodes sufficiently close to the point are found. This operation is made efficient through the use of a search tree data structure. Depending on the type of implementation, the search tree is either an R-tree structure (Guttman, 1984) or a grid-based structure. An R-tree is a bounding volume hierarchy (BVH), i.e. a tree structure of geometrical objects. In the case of the R-tree, the nodes are sorted spatially in a

structure of Cartesian boxes of decreasing size at tree each level. The structure allows for fast neighbor search.

After finding the nearby points, the system (2.14) is solved once for each unique stress component $\boldsymbol{\tau}_{ij,k}$. Since the matrices $\mathbf{M}$ and $\mathbf{B}$ depend only on the location of the Lagrangian nodes included in the interpolation, they are only assembled once.

2.3.4 Coupling to momentum equation

The momentum equation (2.1) is coupled to the viscoelastic constitutive equation through the term $\nabla \cdot \boldsymbol{\tau}$. This term is integrated over the Eulerian control volumes and added to the discretized momentum equation using Gauss's divergence theorem, such that

$$\int_{\Delta V} \nabla \cdot \boldsymbol{\tau} dV = \oint_{\Delta S} \boldsymbol{\tau} \cdot \mathbf{n} dS = \sum_f A_f \mathbf{n}_f \cdot \boldsymbol{\tau}_f, \tag{2.20}$$

where the integral on the left hand side is over the cell volume and is transformed to an integral over the cell surface. For the cell discretization, the surface integral is transformed to a sum over the cell faces. If a face is at the domain boundary or lies inside an immersed boundary, the stress is linearly extrapolated to that face using the value of the neighbor cell in the opposite direction. The configuration of a control volume P with the neighbor cells and faces is shown for two dimensions in Figure 2.4.

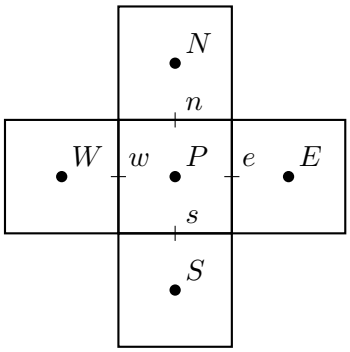

Figure 2.4: Two-dimensional control volume with faces and neighbors.

If BSD is included, i.e. if $\mu_a > 0$, the divergence of the arising explicit term $-2\mu_a \nabla \cdot \mathbf{S}$ (see (1.56)) is integrated in the same way as $\nabla \cdot \boldsymbol{\tau}$ in (2.20). The velocity gradient $\nabla \mathbf{u}$ is calculated using central differences.

2.4 Implementation

The Lagrangian-Eulerian algorithm is implemented for execution on the CPU. In addition, the implicit Euler method to solve the ODE systems and the RBF interpolation are also implemented for simulation on the graphics processing unit (GPU). The different implementations are discussed in this section.

In Table 2.1 a summary of the used algorithms and their respective implementation is given. All CPU code is written in C++ and the GPU code is written in Cuda C++. The Thrust library (Bell & Hoberock, 2011) is used to parallelize the execution of the GPU code.

Routine	CPU-implementation	GPU-implementation
BDF ODE solver	Yes	No
Implicit Euler ODE solver	Yes	Yes
RBF interpolation	Yes	Yes
Node distribution	Yes	No
Stress integration	Yes	No

Table 2.1: Summary of implementations used.

Three different combinations of the implementations listed in Table 2.1 are considered, denoted the CPU-BDF, CPU-Euler and GPU methods, respectively. The methods are summarized in Table 2.2.

Method name	ODE solver	RBF interpolation
CPU-BDF	BDF, CPU	CPU
CPU-Euler	Implicit Euler, CPU	CPU
GPU	Implicit Euler, GPU	GPU

Table 2.2: Combinations of ODE solvers and interpolation methods studied.

2.4.1 ODE solver

The BDF formula is implemented for the CPU using available solvers in the Sundials CVode library (Hindmarsh *et al.*; Sundials, 2020). The implicit Euler method is

implemented for execution both on the CPU and the GPU using in-house solvers.

2.4.2 Unstructured interpolation

The implementation of the unstructured interpolation of the viscoelastic stress involves two main components:

1. Find all Lagrangian nodes close to the point.
2. Solve the system (2.14) and calculate the interpolated value from (2.12) for each stress component.

The first item, i.e. finding the close Lagrangian nodes, is performed slightly differently on the CPU and GPU, respectively. In the CPU implementation, the Lagrangian nodes are stored in an R-tree data structure to find all nodes within a given distance of a point. The R-Tree implementation used in the current work is implemented in the Boost C++ libraries (Dawes & Abraham, 2020). To find the Lagrangian nodes close to a point, all nodes within the distance $\sqrt{d}\Delta x$ are found, where Δx is the local cell size. The factor $\sqrt{d}$ guarantees that all nodes within a Cartesian cell with the side Δx are included in the interpolation.

In the GPU implementation, the Lagrangian nodes are organized in terms of which Cartesian cell they reside in. When finding the Lagrangian nodes close to a cell center, the nodes in that cell are simply included. The reason for choosing a different implementation on for the GPU, even though it could be argued that the R-tree used for the CPU is more general, is that the current grid structure is efficient for the GPU architecture.

Chapter 3

Results and discussion

In this section, applications of the proposed Lagrangian-Eulerian method are presented. The method is first validated by comparing numerical results to the analytic solution for steady and transient planar Poiseuille flow. Simulated results for the flow past a confined cylinder are then compared to numerical data from the literature. The confined cylinder flow is also used to study the computational performance of the proposed CPU and GPU algorithms, respectively. Finally, simulations of a swirl adhesive application are presented and compared to scanned experimental adhesive geometries.

3.1 Planar Poiseuille flow

Two-dimensional planar Poiseuille flows of a single mode upper-convected Maxwell (UCM) with constitutive equation (1.35) and an Oldroyd-B fluid with constitutive equation (1.39) are used to validate the proposed method. A viscosity ratio β is defined as

$$\beta = \frac{\mu}{\mu + \eta}, \tag{3.1}$$

where $\beta = 0$ for an UCM fluid and $0 < \beta < 1$ for an Oldroyd-B fluid. It is noted that β corresponds to the factor $(1 - \lambda_r/\lambda)$ in (1.39).

The computational domain consists of a two-dimensional channel with height $2H$ and length H, where $H = 0.01\,\mathrm{m}$. A schematic representation of the channel is shown in Figure 3.1. Periodic boundary conditions for the velocity are used at the inlet and the outlet. If a Lagrangian node exits through the outlet boundary it is recirculated

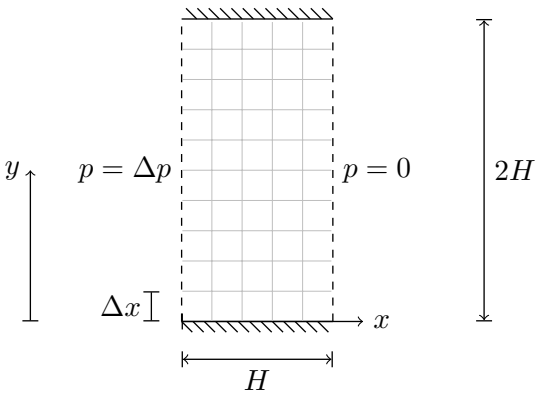

Figure 3.1: Schematic of the planar Poiseulle flow.

through the inlet, and vice versa. The domain thus models an infinitely long channel. The upper and lower walls are modeled as walls with the no-slip condition.

The flow is characterized by $\mathrm{Wi} = \lambda U/H$ and $\mathrm{Re} = \rho HU/\eta$, where U is the mean velocity at fully developed flow. Flow is initiated from rest by imposing a constant pressure drop over the channel. The pressure drop for a given U is calculated from the analytic solution of Waters & King (1970). The accuracy of both transient and fully developed numerical solutions are assessed through comparison with the analytic solution. All results presented in this section have been obtained using the CPU-BDF implementation of the proposed method, see Table 2.2.

For the Lagrangian node distribution $n_{\mathrm{split}} = 2$ and $n_{\max} = 3$ were used. However, it is remarked in these simulations no nodes were added or deleted, which is attributed to the regularity of the flow. The parameter $n_{\max}$ is therefore actually redundant in this case. The Eulerian grid consists of uniform quadratic cells with the side Δx. Different cell sizes Δx are used, which are stated in Table 3.2.

3.1.1 Fully developed flow

The flow is simulated until fully developed for three Weissenberg numbers. Here, $\eta = 1\,\mathrm{Pas}$ and $U = 0.1\,\mathrm{m/s}$ are used and Wi is varied by changing λ. The simulations are repeated for $\beta = 0$, i.e. the UCM fluid, and $\beta = 1/9$. In Table 3.1 the values of λ, Wi and the artificial viscosity μ_a used are listed. The Reynolds number $\mathrm{Re} = 0.001$ for all three flows and the Weissenberg numbers cover two orders of magnitude. It is remarked

that the magnitudes of μ_a used were found suitable to reach the steady solution within reasonable computation time. The large values enable the use of large time steps while maintaining the numerical stability.

λ [s]	Wi	μ_a [Pa s]
0.01	0.1	10^2
0.1	1	10^3
1	10	10^4

Table 3.1: Planar Poiseuille flow parameters.

Spatial accuracy is studied by simulating the flows using uniform grids with different cell sizes Δx. The grids used are defined in Table 3.2. In the simulations where the fully developed solution is of interest, relatively large global time steps, in the range $10^{-3} \leq \Delta t/\lambda \leq 10^{-2}$, are used to reach the fully developed flow as fast as possible. This is feasible since the initial flow transients are not of interest.

Grid	$H/\Delta x$	#cells
G1	5	50
G2	10	200
G3	20	800
G4	40	3200
G5	80	12800

Table 3.2: Grids used in the convergence study for the planar Poiseuille flow.

Since the simulations are transient, fully developed flow is defined with respect to the relative difference

$$\frac{||\phi_n - \phi_{n-1}||}{||\phi_n||} < \varepsilon_{\text{tol}}, \tag{3.2}$$

where ϕ_n is velocity, normal stress or shear stress at global time step n, ε_{tol} is a tolerance and $||\bullet||$ denotes the L_2-norm

$$||\phi_n|| = \sqrt{\sum_{i=1}^{N_{\text{cells}}} \phi_{n,i}^2}, \tag{3.3}$$

where $\phi_{n,i}$ is the property in the ith cell and N_{cells} the number of cells. For the results presented in this section, relative differences in velocity between time steps below $5 \cdot 10^{-13}$ were obtained and relative differences for the stress components were below $5 \cdot 10^{-10}$.

The computed fields are compared across the channel to the analytic profiles for fully developed flow, which read (Xue *et al.*, 2004)

$$u(y) = \frac{3yU}{2H}\left(2 - \frac{y}{H}\right), \tag{3.4}$$

$$\tau_{xx}(y) = \frac{18\lambda\eta U^2}{H^2}\left(1 - \frac{y}{H}\right)^2, \tag{3.5}$$

$$\tau_{xy}(y) = \frac{3\eta U}{H}\left(1 - \frac{y}{H}\right). \tag{3.6}$$

In Figure 3.2 the profiles of velocity, normal stress and shear stress at fully developed flow of the UCM fluid ($\beta = 0$) are shown across the channel at $x = \frac{1}{2}H$ for Wi $= 10$, obtained for grid G1, G3 and G5, Table 3.2. The velocity has been normalized by U and the stresses with their corresponding analytic wall values, $\tau_{xx,w}$ and $\tau_{xy,w}$, respectively. The velocity and stress profiles converge towards the analytic solution for increased grid resolution. The omitted grids follow the same trends.

The corresponding results for Wi $= 0.1$ and Wi $= 1$ are visually identical to those for Wi $= 10$ and have been omitted. Furthermore, the results for the Oldroyd-B model with $\beta = 1/9$ are also identical to those for the UCM fluid and are therefore not subject to further discussion for the fully developed flow.

A quantitative analysis of the convergence rate of the proposed method is performed by computing the error with respect to the analytic solution as

$$E_\phi = \frac{||\phi - \phi_{\text{analytic}}||}{||\phi_{\text{analytic}}||}, \tag{3.7}$$

where ϕ_{analytic} is the analytic velocity, shear stress or normal stress. In Figure 3.3 the computed errors for the simulations of the UCM fluid are shown for different cell sizes. The linear slope in log space is used to estimate the order of accuracy of the method as the spatial resolution $\Delta x \to 0$. A value of 2 is found for velocity and normal stress for the simulated range, indicating second order spatial accuracy.

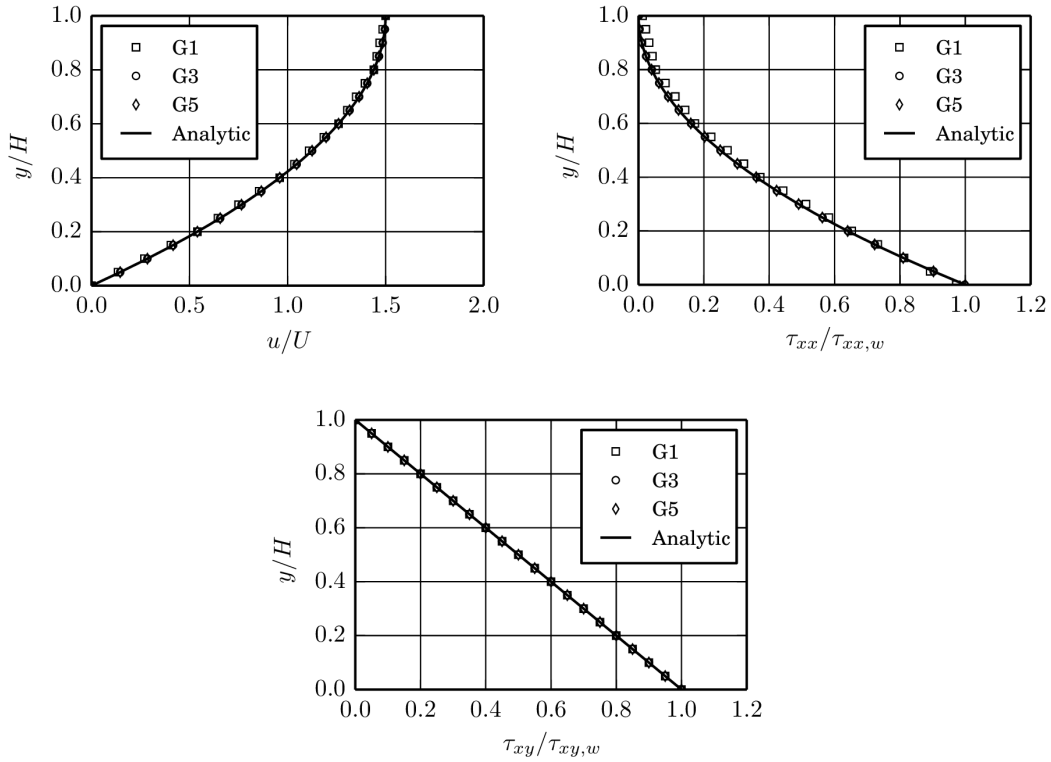

Figure 3.2: Simulated fully developed velocity (top left), shear stress (top right) and normal stress (bottom) calculated with the UCM model for Wi = 10. Computed with $n_{\text{split}} = 2$.

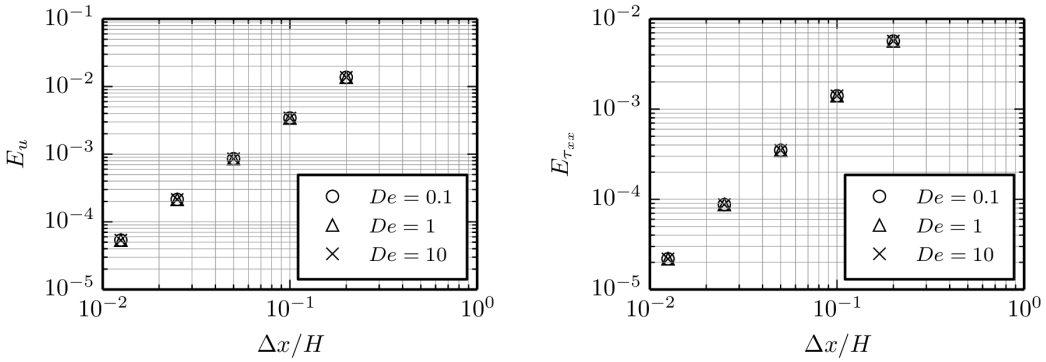

Figure 3.3: Relative errors with respect to analytic for planar Poiseuille flow for velocity (left) and normal stress (right) for Wi = 0.1, Wi = 1 and Wi = 10.

The simulations are repeated with $n_{\text{split}} = 2, 3, 4$ for Wi = 0.1 to study the influence of the relative resolution of the Lagrangian node set. The resulting errors of velocity and normal stress are presented in Figure 3.4. The velocity errors remain unchanged when n_{split} is increased. This is expected since the velocity accuracy is limited by the Eulerian discretization and not by the Lagrangian nodes. The normal stress errors decrease with increasing n_{split}, but the order of accuracy with respect to the grid size Δx remains constant.

3.1.2 Startup flow

In addition to the fully developed solution, the transient startup flow in the channel is studied for different viscosity ratios. As for the fully developed flow, $U = 0.1\,\text{m/s}$ and $\eta = 1\,\text{Pas}$ and Wi is varied through λ. Furthermore, the viscosity ratio β is varied since it has significant effect on the transient flow.

The startup flow is simulated for $\beta = 1/9, 1/18, 1/27$. To ensure a sufficiently small global time step Δt, different step sizes are compared. In Figure 3.5 the velocity at the center of the channel is shown for the first five milliseconds, calculated with three different step sizes for Wi = 0.1 and Wi = 1 with $\beta = 1/27$. The simulations were carried out with the intermediate grid G3 and $n_{\text{split}} = 2$. Furthermore, no BSD is used, i.e. $\mu_a = 0$, to avoid diffusing the solution in time. The predicted velocities converge to

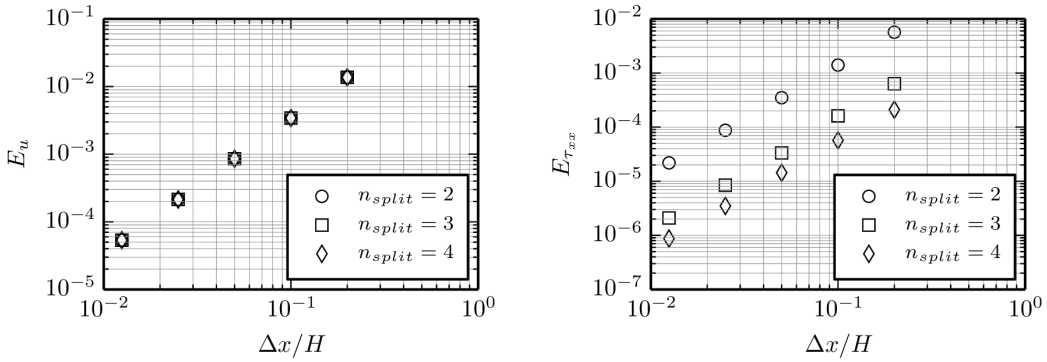

Figure 3.4: Relative errors with respect to analytic for planar Poiseuille flow for velocity (left) and normal stress (right) for $n_{\text{split}} = 2$, $n_{\text{split}} = 3$ and $n_{\text{split}} = 4$.

the analytic solution and the velocities predicted by the two smallest fine steps overlap. It is thus concluded that, for these viscosity ratios, $\Delta t/\lambda = 10^{-3}$ and $\Delta t/\lambda = 10^{-4}$ are sufficiently small for $\text{Wi} = 0.1$ and $\text{Wi} = 1$, respectively. These time step sizes are therefore used for the following simulations.

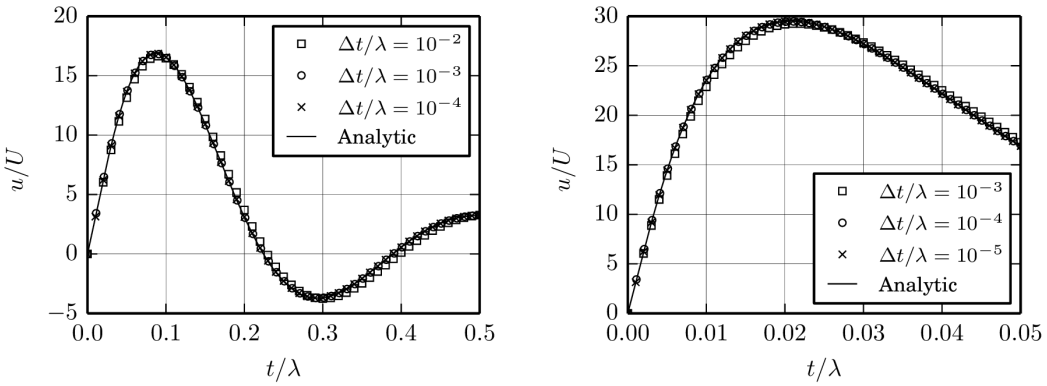

Figure 3.5: Temporal convergence for startup of planar Poiseuille flow for $\text{Wi} = 0.1$ (left) and $\text{Wi} = 1$ (right) with $\beta = 1/27$.

In Figure 3.6 the computed velocities at the channel centerline $y = H$ are shown

for the flows simulated. The computed velocities overlap the analytic solution for the studied range. This indicates that the proposed method correctly predicts the transient viscoelastic flow.

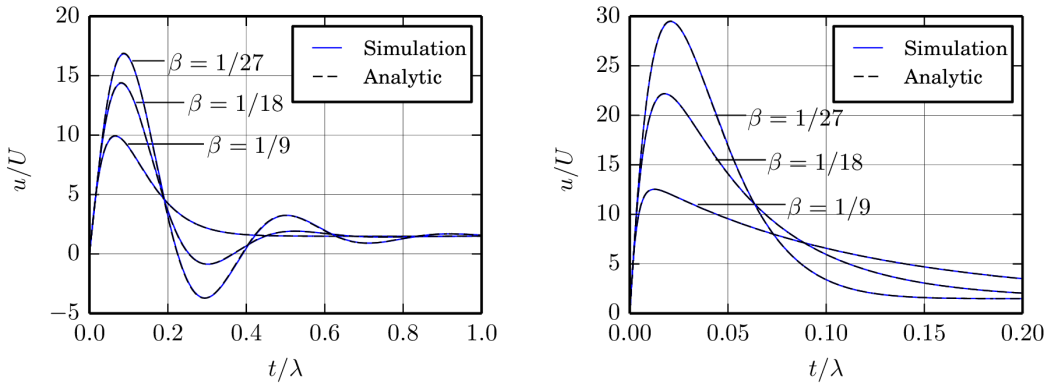

Figure 3.6: Transient centerline velocities for startup of planar Poiseuille flow for Wi = 0.1 (left) and Wi = 1 (right).

Small β and λ result in large velocity overshoot and oscillations. This is demonstrated in Figure 3.6. The performance in terms of accuracy and stability as $\beta \to 0$ is therefore investigated by repeating simulations for $\beta = 0.001$ and $\beta = 0$ with Wi $= 0.1$. For these cases, a small amount of BSD was necessary to maintain numerical stability, and $\mu_a = 0.01\,\text{Pas}$ is therefore used. To reduce the temporal diffusion of the solution, a smaller time step $\Delta t/\lambda = 10^{-4}$ is used. The results are shown in Figure 3.7. For the nonzero $\beta = 0.001$ the simulated velocity resembles the analytic solution excellently. For $\beta = 0$ the simulation predicts the analytic solution fairly well, however with some discrepancies around the peaks of maximum velocity magnitude. The frequency of the oscillations is precisely predicted also for this case.

3.2 Confined cylinder flow

Flow past a confined cylinder is a commonly used benchmark problem for viscoelastic flows simulations. Examples include Alves *et al.* (2001); Baaijens *et al.* (1995); Fraggedakis *et al.* (2016); Hulsen *et al.* (2005); Oliveira *et al.* (1998). Due to the walls

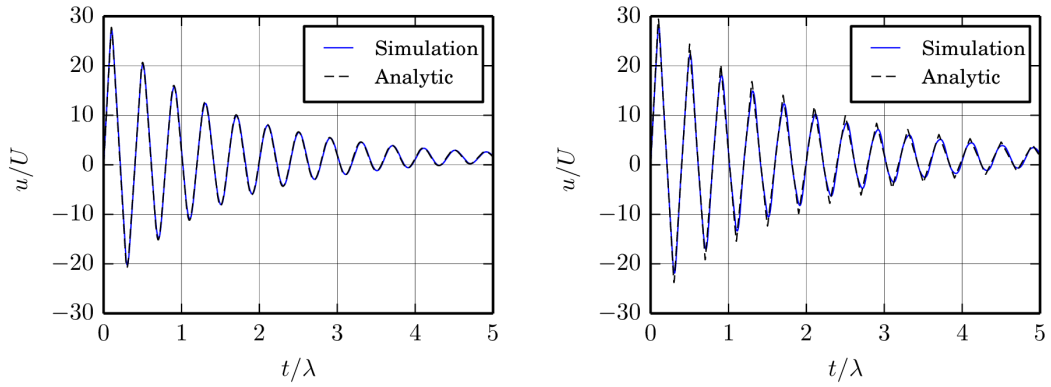

Figure 3.7: Transient centerline velocities for startup of planar Poiseuille flow with $\beta = 0.001$ (left) and $\beta = 0$ (right) for $\mathrm{Wi} = 0.1$.

of the cylinder and the confining channel, the flow exhibits shear and extensional characteristics, respectively, both which have significant effects on viscoelastic flow behavior.

In this work, the flow of a four-mode SPTT fluid is simulated. The linear form of the function $\mathcal{F}$, see (2.2), is used and the constitutive equation for the kth mode reads

$$\lambda \overset{\triangledown}{\boldsymbol{\tau}}_k + \left(1 + \frac{\varepsilon_k \eta_k}{\lambda_k} \mathrm{Tr}\left(\boldsymbol{\tau}_k\right)\right) \boldsymbol{\tau}_k = 2\eta \mathbf{S}, \tag{3.8}$$

where ε_k is a dimensionless parameter. The parameters η_k, λ_k and ε_k for $k = 1, \ldots, 4$ are chosen to match those used by (Baaijens *et al.*, 1995), who performed numerical simulations of the flow using FEM and also recorded the velocity and stress profiles in physical experiments.

Mode	η_k[Pas]	λ_k [s]	ε_k
1	0.443	0.00430	0.39
2	0.440	0.0370	0.39
3	0.0929	0.203	0.39
4	0.00170	3.00	0.39

Table 3.3: Parameters for the SPTT model used in the confined cylinder flow.

The geometry of the two-dimensional channel is shown in Figure 3.8. The cylinder

has radius $R = 2\,\text{mm}$ and is located with its center at $(x/R, y/R) = (0, 0)$, where x and y are the streamwise and the cross-channel direction, respectively. The cylinder wall is modeled by an immersed boundary with the no-slip condition. The height of the channel is $4R$ and the upper and lower boundaries are walls where the no-slip condition is imposed. The total length of the channel is $20R$. At the inlet, which is located at $x/R = -10$, fully developed flow profiles with mean velocity U are imposed. At the outlet, located at $x/R = 10$, zero pressure is imposed and the velocities are extrapolated in the flow direction. The length of the channel was found sufficient for boundary effects not to influence the flow near the cylinder. Uniform grids with cell size $\Delta x = \Delta y$ are used. An example grid has been included in a small part of Figure 3.8 for illustration.

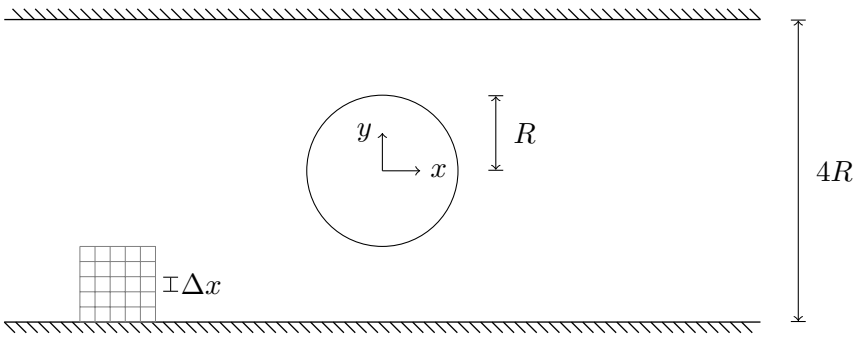

Figure 3.8: Symmetrically confined cylinder geometry.

The total viscosity is defined as

$$\eta_0 = \sum_{k=1}^{4} \eta_k, \tag{3.9}$$

and the mean relaxation time as

$$\bar{\lambda} = \frac{1}{\eta_0} \sum_{k=1}^{4} \lambda_k \eta_k. \tag{3.10}$$

Using these definitions, the Deborah and Reynolds numbers for the flow are calculated as $\text{De} = \bar{\lambda} U/R$ and $\text{Re} = \rho R U/\eta_0$, respectively. Three flow rates are studied, for which the parameters are listed in Table 3.4. The stress quantity $\tau_0 = 3\eta_0 U/R$ is used to normalize stresses when presenting the data. It is noted that the elasticity number

U [m/s]	τ_0[Pa]	De	Re
0.0115	16.9	0.25	0.019
0.0424	62.2	0.93	0.069
0.1074	157.5	2.32	0.174

Table 3.4: Parameters for the simulations of the confined cylinder flow case.

$\mathrm{El} = \mathrm{De}/\mathrm{Re} \approx 13$ for all flow rates, indicating that the flow is dominated by elasticity rather than inertia.

A uniform grid with cell size $\Delta x = \Delta y$ is used. To ensure sufficient spatial resolution, a grid independence study is performed using four grids. The cell sizes of the grids are shown in Table 3.5. The parameters $n_{\text{split}} = 2$ and $n_{\max} = 5$ are used for the Lagrangian node set.

Grid	Δx	N_{cells}
M1	R/10	$8 \cdot 10^3$
M2	R/20	$32 \cdot 10^3$
M3	R/40	$128 \cdot 10^3$
M4	R/60	$288 \cdot 10^3$

Table 3.5: Cell sizes used in the grid independence study for the confined cylinder flow.

Transient flow is simulated and the results are compared for fully developed flow. An adaptive time step is used to make sure that the CFL number is below 0.1 in the whole domain. All results shown in this section have been obtained with the CPU-BDF implementation of the proposed method, see Table 2.2. The computed profiles of velocity, first normal stress difference and shear stress are shown across the channel at $x/R = 1.5$ are shown in Figure 3.9 for the grids defined in Table 3.5.
N_1 is also shown along the channel at $y/R = 0$ downstream of the cylinder. The resolution of grid M1 is clearly too coarse. The comparison across the channel at $x/R = 1.5$ suggests that M2 is sufficient. However, the computed first normal stress differences along the channel reveals that this is not the case. There, the result from M3 is very close to that of M4 on the scale of comparison. It is therefore concluded that grid M3 produces grid independent results.

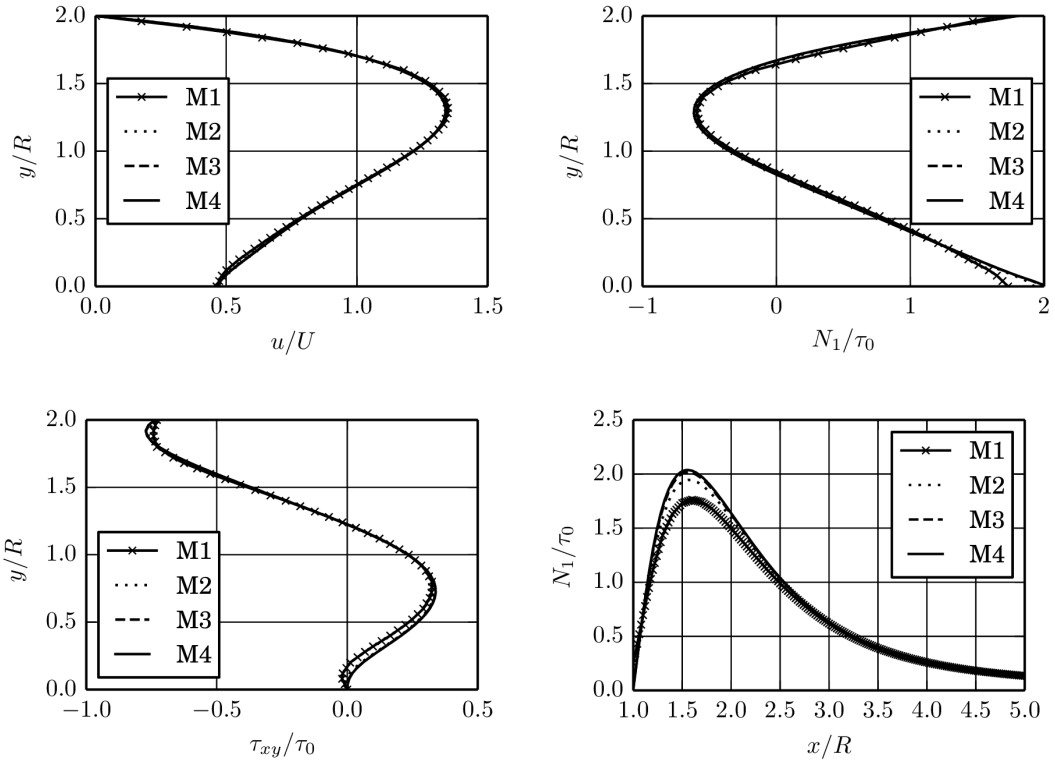

Figure 3.9: Simulated velocity (top left), first normal stress difference (top right) and shear stress (bottom left) across the channel at $x/R = 1.5$ and first normal stress difference along the channel at $y/R = 0$ (bottom right) for the confined cylinder flow with De = 2.32 for the grids defined in Table 3.5.

The two additional implementations of the proposed method defined in Table 2.2, i.e. the CPU-Euler and GPU methods, are validated using grid M3 and compared in the same locations as the grids in Figure 3.9. The results are found to overlap for the three implementations.

An important aspect of the proposed Lagrangian-Eulerian method is how the computed stress fields are affected by the addition and deletion of Lagrangian nodes, particularly near steep stress gradients. To address this, the transient stress during the startup of the simulation is monitored in detail at relevant locations.

Five points are chosen, which are shown in Figure 3.10 and defined in detail in Table 3.6. The three first points, $\mathbf{a}_{\text{wall}}$, $\mathbf{a}_{\text{chan}}$ and $\mathbf{a}_{\text{cyl}}$ are located in the cross section $x/R = 0$. This is the narrowest section of the flow and the fluid is subject to large stress gradients. The remaining two points, $\mathbf{a}_{\text{ups}}$ and $\mathbf{a}_{\text{down}}$, are located where fluid streamlines diverge and converge, respectively. A certain extent of deletion and addition of nodes is therefore expected in these regions.

Point	x/R	y/R
$\mathbf{a}_{\text{wall}}$	0	2
$\mathbf{a}_{\text{chan}}$	0	1.5
$\mathbf{a}_{\text{cyl}}$	0	1
$\mathbf{a}_{\text{ups}}$	-1.5	0
$\mathbf{a}_{\text{down}}$	1.5	0

Table 3.6: Location of points in cylinder channel for monitoring transient stresses.

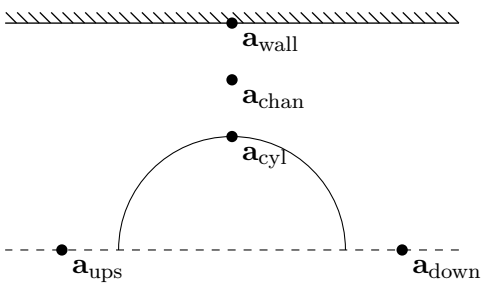

Figure 3.10: Location of points in cylinder channel used to monitor transient stresses.

During the simulation, the number of time steps in which a node is added or deleted,

respectively, within one cell size Δx are recorded for each of the points in Table 3.6 for $0 \leq t/\bar{\lambda} \leq 1/2$. The simulation is performed with grid M3, using $n_{\text{split}} = 2, 3, 4$ and $n_{\max} = 3$.

In Figure 3.11 the recorded frequencies of addition and deletion of nodes, respectively, are shown for the points defined in Table 3.6. Both addition and deletion occur to a certain extent near all the points, with exception for $\mathbf{a}_{\text{wall}}$. There, no deletion is recorded during this time and only a small amount of addition occurs for $n_{\text{split}} = 3, 4$. The highest frequency of addition and deletion is observed around $\mathbf{a}_{\text{ups}}$. A general trend is that the amount of both addition and deletion increases with n_{split}.

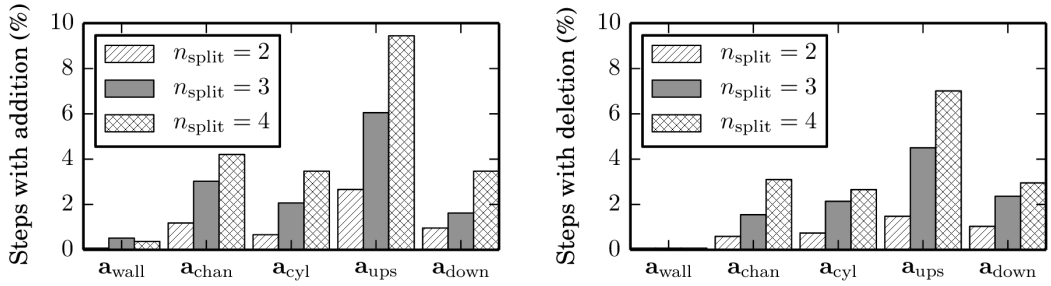

Figure 3.11: Fraction of time steps with node addition (left) and deletion (right) for $0 \leq t/\bar{\lambda} \leq 1/2$ for De $= 2.32$ near the points defined in Table 3.6, using $n_{\text{split}} = 2, 3, 4$ and $n_{\max} = 3$.

In Figure 3.12 the transient variation of N_1 is shown at $\mathbf{a}_{\text{ups}}$ and $\mathbf{a}_{\text{chan}}$, which are locations with high frequency of addition and deletion. The predicted transients obtained with different n_{split} overlap on the scale of comparison. The same result is observed for the remaining points in Table 3.6. The same overlap is also found for the viscoelastic shear stress. The study was furthermore repeated for $n_{\max} = 5$. Both addition and deletion were then less frequent, which is expected since the range of allowed number of nodes in a cell is wider. The transient stresses however overlap those obtained with $n_{\max} = 3$.

Finally, the velocity and stress computed with the proposed method are compared to numerical FEM simulations of Baaijens *et al.* (1995). The experimental results are furthermore included in the comparison as a reference. In Figure 3.13 the velocity,

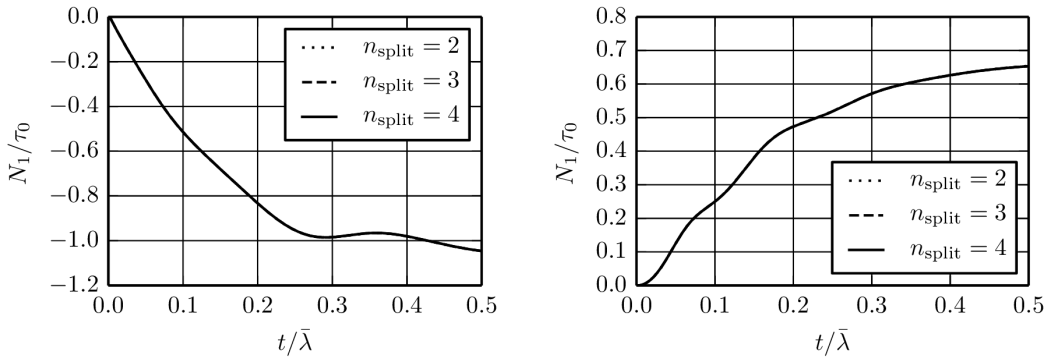

Figure 3.12: Transient first normal stress difference for De = 2.32 at $\mathbf{a}_{\text{ups}}$ (left) and $\mathbf{a}_{\text{chan}}$ (right) and, defined in Table 3.6, for different n_{split} and $n_{\max} = 3$.

first normal stress difference and shear stress are compared across the channel at three locations, $x/R = -5, -1.5, 1.5$. An overall good agreement is found between the simulations with the proposed method and the FEM simulations. Some difference can be seen for N_1 at $x/R = -1.5$. However, the raw data of (Baaijens *et al.*, 1995), which could explain small discrepancies, was not available for the comparison. Furthermore, it is noted that the discrepancies between the two different numerical methods are much smaller than those between numerical and experimental results.

In Figure 3.14 the velocity and first normal computed with the proposed method are compared to the FEM simulations along the channel at $y/R = 0$. Small differences can be seen downstream of the cylinder. However, it is suggested by Figure 3.13 the simulations are in reasonable agreement in this region of the flow. Again, it is noted that the discrepancies between the numerical results are smaller compared to the differences between numerical and experimental data.

To summarize, reasonable agreement is found between the simulations with the proposed model and the FEM simulations of Baaijens *et al.* (1995). Some discrepancies are observed, which may be explained by uncertainties in the data and that the results have been obtained with different computational meshes as well as discretization methods. The observed differences between the two numerical methods being compared is smaller than those between either of the numerical methods and the experimental data.

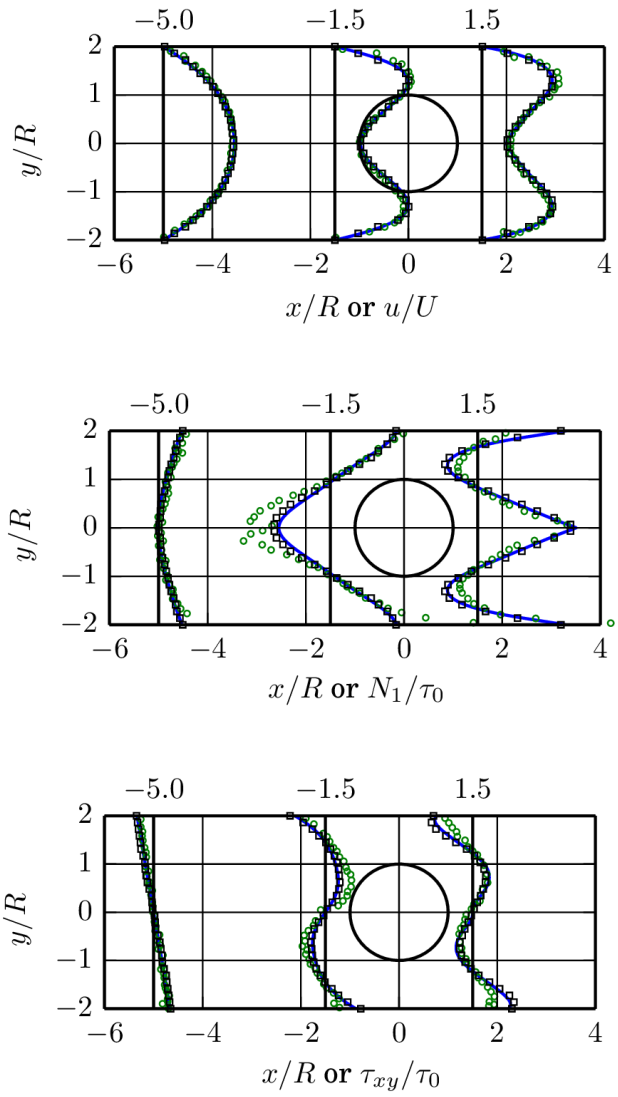

Figure 3.13: Profiles of velocity, u, (top), first normal stress difference, N_1, (middle) and viscoelastic shear stress, τ_{xy}, (bottom) for De = 2.32 across the channel computed with the proposed Lagrangian-Eulerian method (−) compared to FEM-simulations (□) and experiments (○) from Baaijens *et al.* (1995).

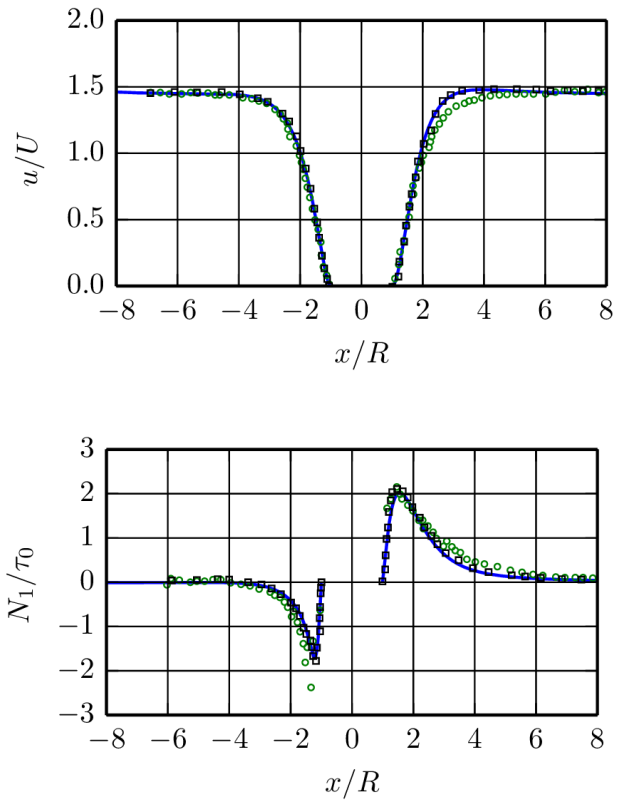

Figure 3.14: Resulting velocity, u, (top) and first normal stress difference, N_1, (bottom) along the channel for De = 2.32 computed with the proposed Lagrangian-Eulerian method (−) compared to FEM-simulations (□) and experiments (○) from Baaijens *et al.* (1995).

3.3 Computational time

The confined cylinder flow of the four-mode PTT fluid discussed in Section 3.2 is used to benchmark the computational performance of the different implementations used, discussed in Section 2.4. Particularly, the comparison of the performance of the GPU algorithm compared to the CPU implementations is of interest.

The flow is simulated for De = 2.32, and the time is measured for different parts of the algorithm and averaged over the first 100 time steps. For this study uniform time steps are used. The calculations are repeated for 1-8 CPU cores for different spatial resolution. The number of Lagrangian nodes ranges from around 10^4 to 10^6, spanning more than two orders of magnitude. All results presented in this section were obtained with an Intel(R) Xeon(R) Gold 6134 CPU with eight 3.20GHz cores and with a Tesla V100 GPU with 32Gb memory.

In Figure 3.15 the average computational times for solving the ODE systems and interpolating the viscoelastic stresses to the Eulerian cell centers, respectively, are shown for the different methods, obtained with four CPU threads for varying spatial resolution. The ODE solution solved with the implicit Euler method is faster than the BDF method both on the CPU and the GPU. Furthermore, the GPU simulation is substantially faster than the CPU simulation. The unstructured interpolation times are the same for both CPU methods, since they run the exact same code. The proposed GPU method is again substantially faster than the CPU method for the interpolation.

In Figure 3.16 the average computational times for solving the ODE systems and interpolating viscoelastic stresses are shown for the different methods and for varying number of processor threads. The results have been obtained with the highest grid resolution used. For both the ODE solution and the interpolation, the times scale with the number of CPU threads for the CPU methods. For the GPU methods the times are practically constant with the number of CPU threads. This attributed to that the parallelization on the GPU is independent of the number of CPU threads. The GPU method is however significantly faster over the range studied.

In Figure 3.17 the average computational times for the full viscoelastic stress calculation and for a full simulation time step are shown for the different methods and for varying number of processor threads. The viscoelastic stress calculation includes the ODE solution, the unstructured interpolation and the node redistribution. Since

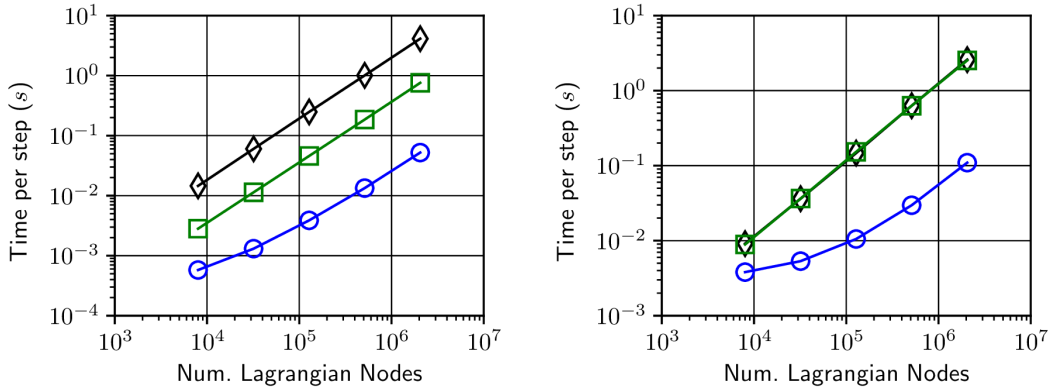

Figure 3.15: Average time for solving ODE system (left) and inteprolating viscoelastic stress to Eulerian cell centers (right) for the four-mode PTT fluid in the confined cylider channel for GPU (○), CPU-BDF (◇) and CPU-Euler (□) using 4 CPU threads.

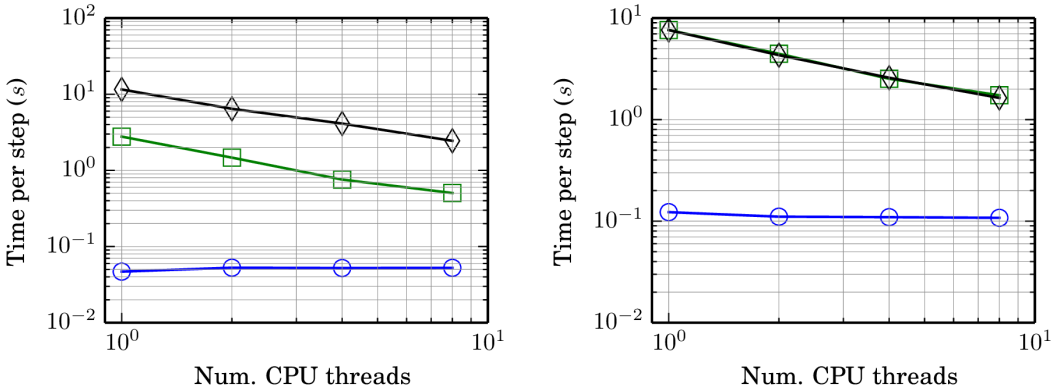

Figure 3.16: Average time for solving the ODE systems (left) and interpolating viscoelastic stresses to Eulerian cell centers (right) for the four-mode PTT fluid in the confined cylider channel for GPU (○), CPU-BDF (◇) and CPU-Euler (□).

the measured times for the GPU method also includes operations performed on the CPU, a slight dependence of number of threads is observed for the viscoelastic stress calculation. As expected, an even stronger dependence is seen for the full time step. The GPU method is faster than the CPU methods for both the stress calculation and the full time step.

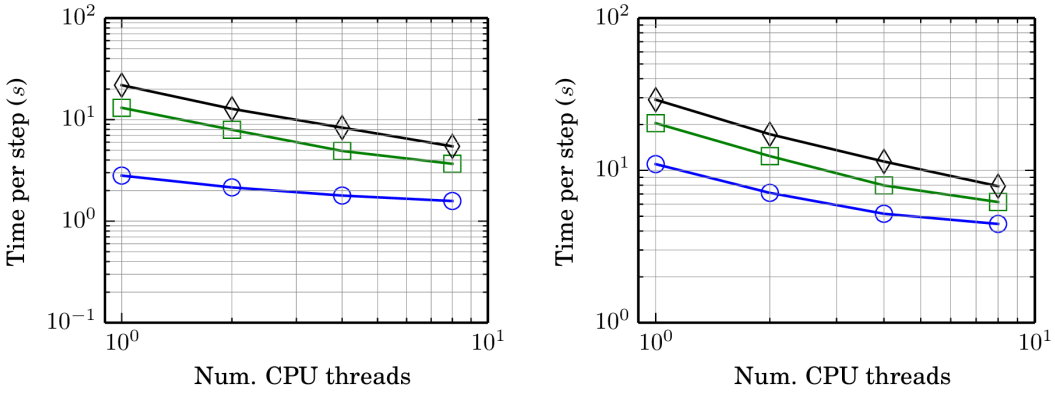

Figure 3.17: Average time for calculating viscoelastic stress (left) and for a full time step (right) for the four-mode PTT fluid in the confined cylider channel for GPU (○), CPU-BDF (◇) and CPU-Euler (□).

Finally, the effect of performing multiple local time steps per global time step is studied for the implicit Euler methods. In Figure 3.18 the average computational times for solving the ODE systems using 10 and 100 local steps. The results for the CPU-BDF method are the same as previously shown, and are included for reference. The time for solving the ODE systems on the CPU is clearly increased when increasing the number of local steps. The corresponding increase is not nearly as pronounced for the GPU solver. This is reasonable since the operation that accounts for the largest portion of the computation time for the GPU method is the memory transfer between the CPU and the GPU. Therefore, taking multiple time steps does not substantially increase the computation time, since the memory is only transferred once per global time step.

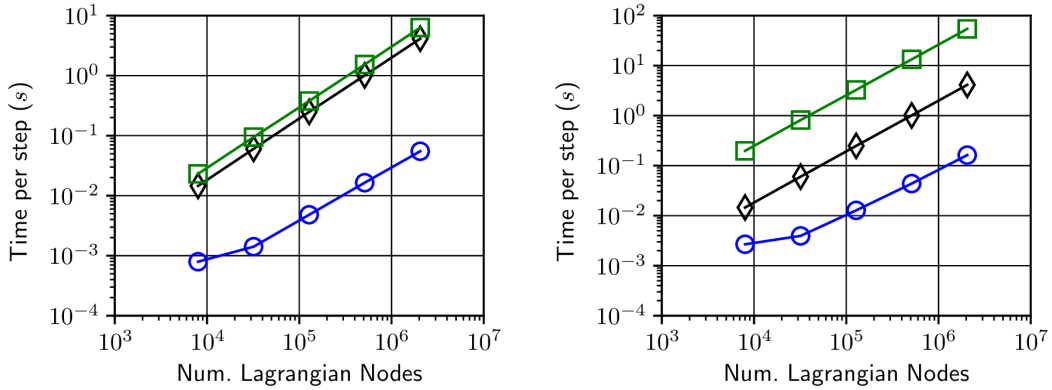

Figure 3.18: Average solving the ODE systems using 10 substeps (left) and 100 substeps (right) for the four-mode PTT fluid in the confined cylider channel for GPU (○) and CPU-Euler (□) using 4 CPU threads. The CPU-BDF (◇) results are included for reference.

3.4 Simulation of deposition applications

A target application for the developed numerical framework is simulation of processes in which viscoelastic fluids are applied onto a geometrical object along a given path. Such applications include for example adhesive application, seam sealing and 3D-printing. The common properties of these flow are

- two-fluid flow of viscoelastic material and a surrounding gas phase with much lower density (i.e. air),
- continuous application of material from an inlet (nozzle) moving along a prescribed path,
- application onto a solid object of arbitrary geometry.

To enable simulation of such flows, a framework to continuously add material flowing from the nozzle has been developed. The framework is denoted the injection model.

3.4.1 Injection modeling framework

The injection model in a simulation has two main steps,

1. Refine the Eulerian octree grid in the area where material should be injected
2. Identify injection cells and add material by modifying the local α and setting the inlet velocity.

The velocity in each injection cell is set to

$$\mathbf{u}_{\text{inj}} = \mathbf{u}_{\text{flow}} + \mathbf{u}_{\text{app}}, \tag{3.11}$$

where $\mathbf{u}_{\text{flow}}$ is the inlet velocity, based on the local volume flow rate and the specific injection model, and $\mathbf{u}_{\text{app}}$ is the velocity of the moving applicator.

A specific distinction can be made between two main categories of injection models. The first is category is direct injection models, in which the injection is performed at the nozzle location and the injection cells represent the geometry of the nozzle. The second category is denoted projected injection models. In this case a simplified model is used to predict how the injected material impacts on the target surface. The material is then injected close to the surface and the free surface flow is simulated. A clear advantage of this strategy is that a large part of the computational cost is removed, as compared to physically resolving the flow from the nozzle to its impact. This approach is of however only suitable for certain applications.

In principle the injection model framework can describe various types of applications at different level of detail. An application is modeled by the construction of a specific injection model. Such an example is discussed in the following section.

3.4.2 Swirl injection model

Swirl application of adhesives is a relatively new technique. The adhesive emerges from a circular nozzle, typically with diameter d_n below one millimeter, positioned at an offset from the axis of application. The nozzle rotates at high speed (10-20krpm), resulting in a spiral-like pattern of the applied adhesive. The advantages compared to traditional cylinder bead application include less sensitivity to deposition geometry variations and that the application can be performed for both higher robot speeds and

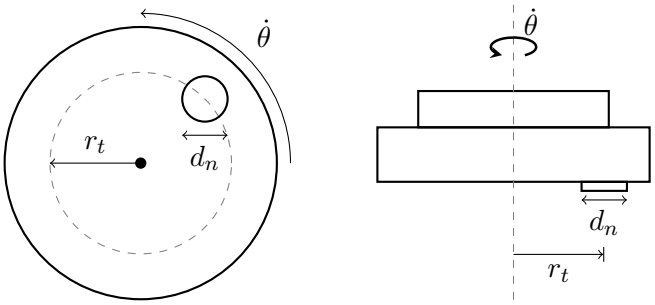

Figure 3.19: Sketch of swirl adhesive nozzle.

at larger distance from the surface. A sketch of the swirl application concept is given in Figure 3.19.

The flow of adhesive from the nozzle is characterized by high velocity, small nozzle diameter and high rotation speeds. Consequently, very high spatial and temporal resolutions are needed to resolve the flow in detail. In this work, a process-compatible injection model is instead proposed. The flow from the nozzle to the surface is modeled instead of being fully resolved. The material is thus injected close to the target surface. This model significantly reduces the computational cost to simulate the process, such that industrial cases may be simulated within reasonable time.

The model is based two main assumptions. Firstly, it is assumed that the momentum of the adhesive leaving the nozzle is large compared to that of the surrounding air. Secondly, the velocity of the adhesive in the application direction is assumed to be sufficiently large for gravity effects to be negligible before the adhesive impacts on the object surface. Based on these assumptions, the adhesive travel from the nozzle to the object surface can be modeled.

The injection model used for the swirl application is based on a torus geometry. In each time step, the impact location of the adhesive if calculated using ray-tracing. The adhesive is then injected to the simulation domain in torus segments calculated from a given angular velocity $\dot{\theta}$, a torus radius r_t and the nozzle diameter d_n. In time step i, with length Δt_i, the angle swept assuming constant angular velocity is $\Delta\theta_i = \dot{\theta}\Delta t_i$. Material is thus injected in the torus segment defined by the angles $\theta \in [\theta_{i-1}, \theta_i]$, where $\theta_i = \theta_{i-1} + \Delta\theta$. The torus segment injection concept is shown in Figure 3.20.

The thickness of the torus is taken as the nozzle diameter d_n. The torus radius r_t is

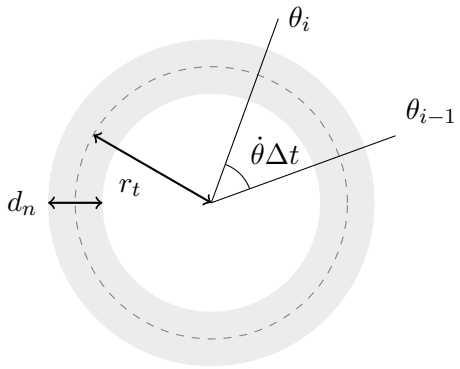

Figure 3.20: Sketch of time-dependent torus segment.

generally unknown, but depends on the adhesive volume flow rate, the rotation speed and the application distance, i.e. the distance to the deposition geometry. Then

$$r_t = r_t(\dot{\theta}, \dot{V}, L_{\text{app}}), \tag{3.12}$$

where $\dot{V}$ is the volume flow rate and L_{app} the application distance. An approximation of the function (3.12) is obtained by measuring r_t for different process conditions and store it in a database. Whenever r_t is required in the simulation it is interpolated from the database based on the current process conditions. The swirl injection model can be summarized by the following steps:

1. find the location of impact close to the surface using ray-tracing,
2. calculate the torus radius r_t,
3. calculate the torus segment parameters,
4. refine the computational grid in the injection region,
5. identify injection cells and inject material.

3.4.3 Swirl adhesive simulations

The proposed injection model framework is demonstrated for the swirl adhesive application. A set of 3D point clouds scanned from experimental swirl geometries are available, which are used to measure the widths and construct an approximation for

the function (3.12) for the torus radius r_t. The adhesive was applied onto a flat plate by an applicator moving along straight lines with constant velocity 300 mm/s and application distance of $L_{\text{app}} = 30\,\text{mm}$. The nozzle diameter is 0.6 mm. The flow rates and rotation speeds used are listed in Table 3.7. Experiments performed to obtain the 3D point clouds were performed at RISE IVF in Mölndal, Sweden, and the resulting scanned adhesive geometries are shown in Figure 3.21. It is observed that low flow rate and high rotation speed, as for setup C, results in a narrow and dense adhesive pattern. High flow rate and low rotation speed, as for setup D, results in a wide and sparse pattern.

Setup	Rotation speed [rpm]	flow rate [ml/s]
A	10000	1.14
B	10000	1.7
C	20000	1.14
D	20000	1.7

Table 3.7: Process conditions used for the available swirl application experiments. The application distance $L_{\text{app}} = 30\,\text{mm}$ and robot end-effector speed 300 mm/s is used for all setups.

The application is simulated for the four setups defined in Table 3.7. The simulation domain is a Cartesian box with the sides 50 mm, 100 mm and 50 mm respectively in the x, y and z directions. The coarsest cells are cubes with the side 1 mm. Refinements are generated by recursively splitting cells into eight smaller cubes. Around the viscoelastic adhesive the grid is refined 6 times, resulting in in cells with the size $1/2^6\,\text{mm} \approx 0.016\,\text{mm}$. A constant time step $\Delta t = 5 \cdot 10^{-4}\,\text{s}$ is used. This corresponds to 6 time steps per revolution for 20 krpm and 12 steps per revolution for 10 krpm.

The adhesive is modeled as a single-mode SPTT model with the linear form of the relaxation function, i.e. the constitutive equation in the form of (1.40), with $\lambda = 0.082\,\text{s}$, $\eta = 3065\,\text{Pas}$ and $\varepsilon = 0.5$. The Lagrangian node set has the parameters $n_{\text{split}} = 2$ and $n_{\max} = 3$. It is remarked that the real adhesive likely requires multiple relaxation modes in order to correctly capture its flow behavior. However, the purpose of these simulations is firsthand to evaluate the performance of the injection model. The detailed

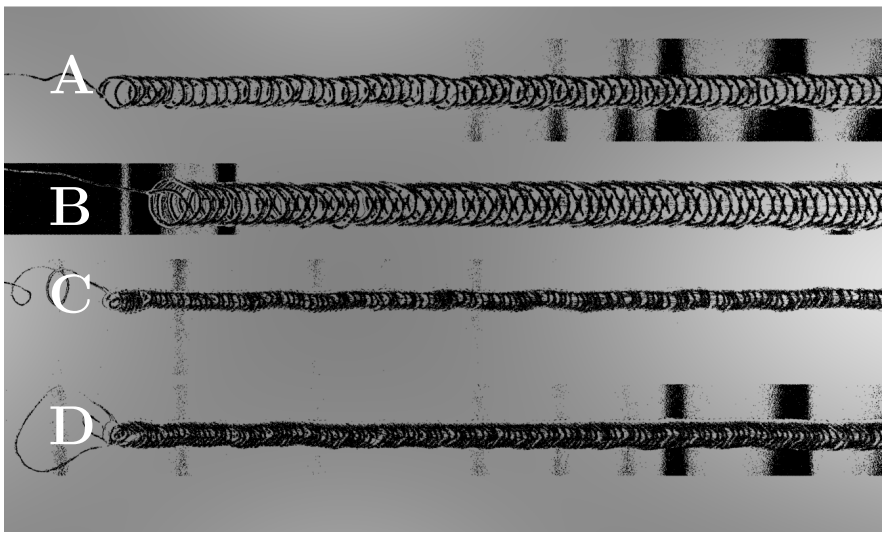

Figure 3.21: Point clouds of scanned experimental swirl adhesive beads corresponding to setups A through D.

assessment of the viscoelastic stress fields for free surface flows is subject to future research.

In Figure 3.22 a snapshot from the simulation with setup B is shown. The adhesive is visualized by the contour surface $\alpha = 0.5$ and the injection cells as solid cubes. As a result of the torus segment injection geometry, the adhesive geometry resembles the spiral-like pattern seen in the corresponding experiments.

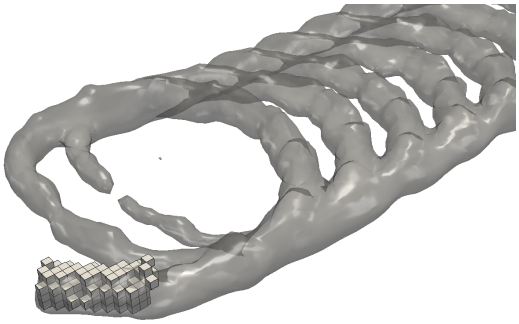

Figure 3.22: Swirl simulation bead geometry visualized as the contour surface $\alpha = 0.5$ and injection cells (solid cubes).

It may be argued that the spatial and temporal resolution are somewhat coarse. For example, the uneven shape or "knuckles", which can be seen in Figure 3.22, may be attributed to the spatial resolution. However, the intent is to propose a numerical model which is suitable for industrially relevant flows. An extremely important property for simulations to be useful in such a context is high computation speed. There is therefore a distinct trade-off between the speed and the accuracy of the simulation. The qualitative accuracy of the simulation model is therefore assessed by comparison with the experimental data.

The simulated and experimental swirl adhesive geometries are compared for setup A and setup B in Figure 3.23 and for setup C and setup D in Figure 3.24. For the low rotation speed (Figure 3.23) the simulated adhesive geometry clearly resembles the experimental adhesive geometry. For the high rotation speed (Figure 3.24), the frequency of the spiral is higher, resulting in the denser pattern. In the simulation, the gaps seen in the experiment are not visible, while the frequency of the spiral pattern is predicted and is in good agreement with the experiments. This discrepancy is, at least partly, attributed to the resolution of the simulation. Since the high rotation speed causes the pattern to be denser, higher spatial resolution would be required to distinguish the gaps more clearly. However, it is also conceivable that shadows in the scanning of the experimental point cloud reveal the gaps as larger than in reality. This is a potential source of error. Overall, good qualitative agreement between the simulated and experimental adhesive geometries is found.

To summarize, the numerical simulations with the swirl injection model and the comparison with experimental data is a promising step towards an industrially viable model. Furthermore, they are an important step towards the ability to simulate the full mechanical joining process, including adhesive application as well as parts assembly and hemming with adhesives.

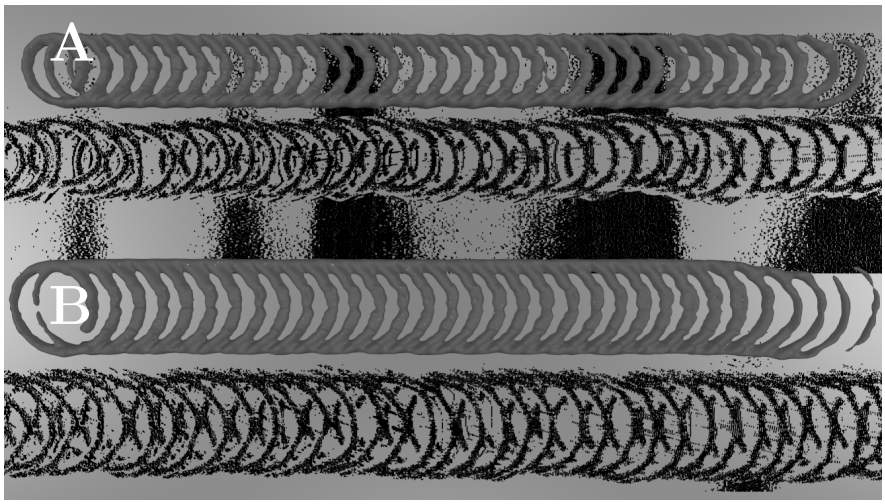

Figure 3.23: Comparison between simulated and experimental swirled adhesive beads for setup A and setup B.

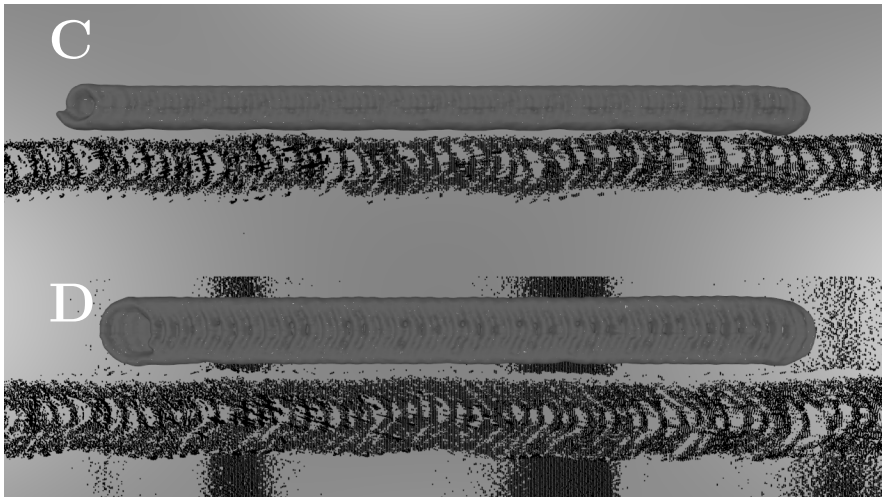

Figure 3.24: Comparison between simulated and experimental swirled adhesive beads for setup C and setup D.

Chapter 4

Conclusions

Viscoelastic fluids appear in many industrial processes and new numerical methods are required for numerical simulations to be feasible, both in terms of simulation time and manual preparation complexity. In this thesis, a new Lagrangian-Eulerian framework for numerical simulations of transient viscoelastic fluid flow has been proposed. The constitutive equation for viscoelastic stress is solved along the trajectories of Lagrangian nodes convected by the flow. The fluid momentum and continuity equations are solved with the finite volume method on an adaptive octree mesh. The viscoelastic stress is interpolated to the Eulerian grid using radial basis functions. The proposed method is implemented for pure CPU simulation as well as in a GPU-accelerated version. Compared to other Lagrangian or Lagrangian-Eulerian methods, no expensive re-meshing due to mesh deformation is needed and a relatively small amount of Lagrangian nodes is sufficient for accurate and stable simulations. Furthermore, no other stabilization approach than both sides diffusion was found necessary for the flows studied in this work.

In Chapter 2 the proposed numerical method was described in detail and the different implementations of the method were discussed. In Chapter 3 the performance of the method was assessed both in terms of accuracy and computational efficiency. Numerical benchmarks showed that the results were in good agreement with analytic solutions as well as numerical and experimental data from the literature. It was also found that a substantial reduction of computational time was obtained by using the GPU-accelerated implementation of the method.

Finally, an applied framework for modeling application of viscoelastic fluids along a prescribed path onto a product geometry was proposed and demonstrated for a swirl

adhesive application. The simulated adhesive geometries were found to be in good qualitative agreement with scanned experimental geometries.

The results presented in this thesis have shown that there is good potential for the proposed method to be used efficiently for real-life industrial scale viscoelastic flow problems:

- The validation with analytic solutions and numerical data from the literature shows that sufficient accuracy can be achieved.
- The computational performance study demonstrates that the method is suitable for GPU-acceleration with substantial decrease in simulation time as a result.
- The swirl adhesive case demonstrates that the proposed Lagrangian-Eulerian method, in combination with suitable process models, can be used for simulation complex problems which are industrially relevant.

The proposed framework is a promising step towards the ability to simulate industrial applications that involve complex flow of viscoelastic fluids. In future work, the proposed method will be evaluated and validated in detail for viscoelastic free flows with the volume of fluids method. This will be done for relevant numerical benchmark problems as well as for industrially relevant flows. Furthermore, it should be evaluated what possible extensions of the method are feasible to improve the method, including its accuracy, computational performance or stability.

Summary of papers

Paper I - A Lagrangian-Eulerian framework for simulation of transient viscoelastic fluid flow

Authors: S. Ingelsten, A. Mark, F. Edelvik.

Journal of Non-Newtonian fluid mechanics

This journal article features the presentation of the Lagrangian-Eulerian method for simulation of viscoelastic fluid flow that this thesis builds upon. The method is validated by comparing simulated flow quantities to analytic solutions for a steady and a transient pressure-driven channel flow of an upper-convected Maxwell (UCM) fluid. The method is also compared to numerical results for the flow of a four-mode Phan Thien Tanner (PTT) fluid past a confined cylinder in a channel.

Paper II - Computationally efficient viscoelastic flow simulation using a Lagrangian-Eulerian method and GPU-acceleration

Authors: S. Ingelsten, A. Mark, K. Jareteg, R. Kádár, F. Edelvik.

Journal of Non-Newtonian fluid mechanics

This paper was submitted to the special issue of JNNFM following the 19th International Workshop on Numerical Methods for Non-Newtonian Flows (IWNMNNF), held in Peso da Régua, Portugal on June 16-20, 2019. In the paper it is described how substantial parts of the Lagrangian-Eulerian method presented in **Paper I** can be implemented for execution on the graphic processing unit (GPU). The resulting impact on the computational speed is also studied.

Paper III - A numerical framework for simulation of swirled adhesive application

Authors: S. Ingelsten, A. Mark, R. Kádár, F. Edelvik.

Annual transactions of the Nordic rheology society

This paper was presented at the Nordic Rheology Conference, held in Gothenburg, Sweden on August 21-23, 2019. In the paper a numerical framework to simulate swirled adhesive application for process scale problems is introduced. Simulated adhesive geometries are compared to scanned experimental results.

Contribution report

Paper I Main author and proposed the idea to use the Lagrangian-Eulerian method for the viscoelastic constitutive equation. Responsible for the implementation of the method into the in-house finite volume flow solver and for running the simulations.

Paper II Main author and proposed the idea to implement the ode solver and the unstructured interpolation on the GPU. Responsible for the implementation of the GPU algorithm and for running the simulations.

Paper III Main author and proposed the idea for the torus section injection geometry to model the swirl within the existing injection model framework. Responsible for the implementation and for running the simulations.

References

Alves, M., Pinho, F. & Oliveira, P. (2001). The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *Journal of Non-Newtonian Fluid Mechanics*, **97**, 207 – 232. 15, 40

Alves, M.A., Oliveira, P.J. & Pinho, F.T. (2003). Benchmark solutions for the flow of Oldroyd-B and PTT fluids in planar contractions. *Journal of Non-Newtonian Fluid Mechanics*, **110**, 45 – 75. 15

Baaijens, H.P., Peters, G.W., Baaijens, F.P. & Meijer, H.E. (1995). Viscoelastic flow past a confined cylinder of a polyisobutylene solution. *Journal of Rheology*, **39**, 1243 – 1277. 15, 40, 41, 46, 47, 48, 49

Balci, N., Thomases, B., Renardy, M. & Doering, C.R. (2011). Symmetric factorization of the conformation tensor in viscoelastic fluid models. *Journal of Non-Newtonian Fluid Mechanics*, **166**, 546 – 553, xVIth International Workshop on Numerical Methods for Non-Newtonian Flows. 19

Barnes, H.A., Hutton, J.F. & Walters, K. (1989). *An Introduction to Rheology*. Rheology Series, Vol. 3 Elsevier, New York. 3, 5, 8, 10

Bell, N. & Hoberock, J. (2011). Thrust: A productivity-oriented library for cuda. In W. mei W. Hwu, ed., *GPU Computing Gems*, chap. 26, 359–371. 31

Bird, R., Curtiss, C., Armstrong, R. & Hassager, O. (1987a). *Dynamics of polymeric liquids Vol. 2 Kinetic Theory*. John Wiley and Sons Inc., New York, NY. 12

Bird, R.B., Armstrong, R. & Hassager, O. (1987b). *Dynamics of polymeric liquids Vol. 1: Fluid mechanics*, vol. 1. John Wiley and Sons Inc., New York, NY, 2nd edn. 3, 9, 11, 12, 13

Chen, X., Marschall, H., Schäfer, M. & Bothe, D. (2013). A comparison of stabilisation approaches for finite-volume simulation of viscoelastic fluid flow. *International Journal of Computational Fluid Dynamics*, **27**, 229–250. 14, 20

Chilcott, M. & Rallison, J. (1988). Creeping flow of dilute polymer solutions past cylinders and spheres. *Journal of Non-Newtonian Fluid Mechanics*, **29**, 381 – 432. 14

Dawes, B. & Abraham, D. (2020). Boost C++ libraries, https://www.boost.org/. 32

Doormaal, J.P.V. & Raithby, G.D. (1984). Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*, **7**, 147–163. 24

Edelvik, F., Mark, A., Karlsson, N., Johnson, T. & Carlson, J. (2017). Math-based algorithms and software for virtual product realization implemented in automotive paint shops. In L. Ghezzi, D. Hömberg & C. Landry, eds., *Math for the Digital Factory*, 231–251, Springer-Verlag, Berlin. 21

Fattal, R. & Kupferman, R. (2004). Constitutive laws for the matrix-logarithm of the conformation tensor. *Journal of Non-Newtonian Fluid Mechanics*, **123**, 281 – 285. 18, 19

Fattal, R. & Kupferman, R. (2005). Time-dependent simulation of viscoelastic flows at high weissenberg number using the log-conformation representation. *Journal of Non-Newtonian Fluid Mechanics*, **126**, 23 – 37. 19

Fraggedakis, D., Dimakopoulos, Y. & Tsamopoulos, J. (2016). Yielding the yield-stress analysis: a study focused on the effects of elasticity on the settling of a single spherical particle in simple yield-stress fluids. *Soft Matter*, **12**, 5378–5401. 40

Gallez, X., Halin, P., Lielens, G., Keunings, R. & Legat, V. (1999). The adaptive lagrangian particle method for macroscopic and micro–macro computations

of time-dependent viscoelastic flows. *Computer Methods in Applied Mechanics and Engineering*, **180**, 345 – 364. 16, 27

Göhl, J., Markstedt, K., Mark, A., Håkansson, K., Gatenholm, P. & Edelvik, F. (2018). Simulations of 3d bioprinting: predicting bioprintability of nanofibrillar inks. *Biofabriaction*, **10**. 21

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, 47–57, Association for Computing Machinery, New York, NY, USA. 29

Halin, P., Lielens, G., Keunings, R. & Legat, V. (1998). The lagrangian particle method for macroscopic and micro–macro viscoelastic flow computations dedicated to professor marcel j. crochet on the occasion of his 60th birthday.1. *Journal of Non-Newtonian Fluid Mechanics*, **79**, 387 – 403. 15

Harlen, O., Rallison, J. & Szabo, P. (1995). A split lagrangian-eulerian method for simulating transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, **60**, 81 – 104. 15

Herrchen, M. & Öttinger, H.C. (1997). A detailed comparison of various FENE dumbbell models. *Journal of Non-Newtonian Fluid Mechanics*, **68**, 17 – 42. 13

Hindmarsh, A.C., Serban, R. & Reynolds, D.R. (????). *User Documentation for cvode v3.1.0 (sundials v3.1.0)*. Sundials. 28, 31

Hulsen, M.A., Fattal, R. & Kupferman, R. (2005). Flow of viscoelastic fluids past a cylinder at high weissenberg number: Stabilized simulations using matrix logarithms. *Journal of Non-Newtonian Fluid Mechanics*, **127**, 27 – 39. 15, 40

Ingelsten, S., Mark, A. & Edelvik, F. (2019). A Lagrangian-Eulerian framework for simulation of transient viscoelastic fluid flow. *Journal of Non-Newtonian Fluid Mechanics*, **266**, 20 – 32. 20

Iske, A. (2004). *Multiresolution Methods in Scattered Data Modelling*, vol. 37 of *Lecture notes in computational science and engineering*. Springer, 1st edn. 25, 29

KEUNINGS, R. (2000). A survey of computational rheology. In *Proceedings of the XIIIth International Congress on Rheology*, vol. 1, 7–14, Citeseer. 18

LARSON, R. (1999). *The Structure and Rheology of Complex Fluids*. Topics in Chemical Engineering, OUP USA. 10

LARSON, R.G. (1988). *Constitutive Equations for Polymer Melts and Solutions*. Butterworths series in chemical engineering, Butterworth Publishers. 9, 11

LIKHTMAN, A.E. & GRAHAM, R.S. (2003). Simple constitutive equation for linear polymer melts derived from molecular theory: Rolie–poly equation. *Journal of Non-Newtonian Fluid Mechanics*, **114**, 1 – 12. 15

LODGE, A. (1974). *Body Tensor Fields in Continuum Mechanics: With Applications to Polymer Rheology*. Academic Press. 8

MARK, A. & VAN WACHEM, B.G.M. (2008). Derivation and validation of a novel implicit second-order accurate immersed boundary method. *J. of Comput. Physics*, **227**, 6660 – 6680. 21, 24

MARK, A., RUNDQVIST, R. & EDELVIK, F. (2011). Comparison between different immersed boundary conditions for simulation of complex fluid flows. *Fluid dynamics & materials processing*, **7**, 241–258. 21, 24

MARK, A., SVENNING, E. & EDELVIK, F. (2013). An immersed boundary method for simulation of flow with heat transfer. *International Journal of Heat and Mass Transfer*, **56**, 424 – 435. 21

MARK, A., BOHLIN, R., SEGERDAHL, D., EDELVIK, F. & CARLSON, J.S. (2014). Optimisation of robotised sealing stations in paint shops by process simulation and automatic path planning. *International Journal of Manufacturing Research 5*, **9**, 4–26. 21

MCLEISH, T.C.B. & LARSON, R.G. (1998). Molecular constitutive equations for a class of branched polymers: The pom-pom polymer. *Journal of Rheology*, **42**, 81–110. 15

Mitsoulis, E. (2013). 50 years of the k-bkz constitutive relation for polymers. *ISRN Polymer Science*. 15

Morozov, A. & Spagnolie, S.E. (2015). *Introduction to Complex Fluids*, 3–52. Springer New York, New York, NY. 8, 12, 14

Oldroyd, J. (1950). On the formulation of rheological equations of state. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 200, 523–541, The Royal Society. 9

Oliveira, P., Pinho, F. & Pinto, G. (1998). Numerical simulation of non-linear elastic flows with a general collocated finite-volume method. *Journal of Non-Newtonian Fluid Mechanics*, **79**, 1 – 43. 15, 40

Owens, R. & Phillips, T. (2002). *Computational Rheology*. Computational Rheology, Imperial College Press. 18

Palhares Junior, I.L., Oishi, C.M., Afonso, A.M., Alves, M.A. & Pinho, F.T. (2016). Numerical study of the square-root conformation tensor formulation for confined and free-surface viscoelastic fluid flows. *Advanced Modeling and Simulation in Engineering Sciences*, **3**, 2. 20

Patankar, S.V. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation. 24

Poole, R.J. (2012). The Deborah and Weissenberg numbers. In *Rheology bulletin*, vol. 53, 32 – 39, British Society of Rheology. 17, 18

Rasmussen, H. & Hassager, O. (1995). Simulation of transient viscoelastic flow with second order time integration. *Journal of Non-Newtonian Fluid Mechanics*, **56**, 65 – 84. 15

Reiner, M. (1964). The Deborah number. *Physics*, **17**, 62. 16

Schlichting, H. & Gersten, K. (2000). *Boundary Layer Theory*. Springer-Verlag Berlin Heidelberg, 8th edn. 6, 7, 16

Stewart, P.A., Lay, N., Sussman, M. & Ohta, M. (2008). An improved sharp interface method for viscoelastic and viscous two-phase flows. *Journal of Scientific Computing*, **35**, 43–61. 20

Sundials (2020). Suite of nonlinear and differential/algebraic equation solvers cvode, http://computation.llnl.gov/projects/sundials/cvode. 31

Svenning, E., Mark, A. & Edelvik, F. (2014). Simulation of a highly elastic structure interacting with a two-phase flow. *Journal of Mathematics in Industry*, **4**, 7. 21

Svensson, M., Mark, A., Edelvik, F., Kressin, J., Bohlin, R., Segerdahl, D., Carlson, J.S., Wahlborg, P.J. & Sundbäck, M. (2016). Process simulation and automatic path planning of adhesive joining. *Procedia CIRP*, **44**, 298 – 303, 6th CIRP Conference on Assembly Technologies and Systems (CATS). 21

Thien, N.P. & Tanner, R.I. (1977). A new constitutive equation derived from network theory. *Journal of Non-Newtonian Fluid Mechanics*, **2**, 353–365. 12

Truesdell, C. & Rajagopal, K. (1999). *An Introduction to the Mechanics of Fluids*. Birkhäuser Boston. 6, 7

Tryggvason, G., Scardovelli, R. & Zaleski, S. (2011). *Direct Numerical Simulations of Gas-Liquid Multiphase Flows.*. Cambridge University Press. 25

Ubbink, O. & Issa, R. (1999). A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics*, **153**, 26 – 50. 25

IPS IBOFlow (2020). IPS IBOFlow, http://ipsiboflow.com. 21

Wapperom, P., Keunings, R. & Legat, V. (2000). The backward-tracking lagrangian particle method for transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, **91**, 273 – 295. 16

Waters, N.D. & King, M.J. (1970). Unsteady flow of an elastico-viscous liquid. *Rheologica Acta*, **9**, 345–355. 34

White, J.L. (1964). Dynamics of viscoelastic fluids, melt fracture, and the rheology of fiber spinning. *Journal of Applied Polymer Science*, **8**, 2339–2357. 17

Xue, S.C., Tanner, R. & Phan-Thien, N. (2004). Numerical modelling of transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, **123**, 33 – 58. 19, 36